INVESTIGATION OF GENETIC ALGORITHM DESIGN REPRESENTATION FOR MULTI-OBJECTIVE TRUSS OPTIMIZATION

A Thesis

by

SOUMYA SUNDAR PATHI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2006

Major Subject: Civil Engineering

INVESTIGATION OF GENETIC ALGORITHM DESIGN REPRESENTATION FOR MULTI-OBJECTIVE TRUSS OPTIMIZATION

A Thesis

by

SOUMYA SUNDAR PATHI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Anne Raich
Committee Members,	Joseph Bracci
	Michael Anshelevich
Head of Department,	David Rosowsky

August 2006

Major Subject: Civil Engineering

ABSTRACT

Investigation of Genetic Algorithm Design Representation for
Multi-Objective Truss Optimization. (August 2006)

Soumya Sundar Pathi, B.Tech., National Institute of Technology Calicut, India
Chair of Advisory Committee: Dr. Anne Raich

The objective of this research is to develop a flexible design grammar and genetic algorithm representation to be used in a multi-objective optimization method to design efficient steel roof trusses given space dimensions and loading requirements by the user. The goal of implementing the method as a multi-objective problem is to obtain a set of near-optimal trusses for the defined unstructured problem domain, not just a single near-optimal design. The method developed was required to support the exploration of a broad range of conceptual designs before making design decisions. Therefore, a method was developed that could define numerous design variables, support techniques to locate global or near-global optimal designs, and improve the efficiency of the computational procedures implemented. This research effort was motivated by the need to consider structural designs that may be beyond the established conventions of designers in the search for cost-efficient, structurally-sound designs.

An effective design grammar that is capable of generating stable trusses is defined in this research. The design grammar supports the optimization of member size, in addition to truss geometry and topology. Multi-objective genetic algorithms were used to evolve sets of Pareto-optimal trusses that had varying topology, geometry, and member sizes. The Pareto-optimal curves provided design engineers with a range of near-optimal design alternatives that showed the tradeoffs that occur in meeting the stated objectives. Designers can select their final design from this set based on their own individual weighting of the design objectives. Trials are performed using a multi-objective genetic algorithm that works with the design grammar to evolve trusses for

different span lengths. In addition to evaluate the performance of the developed optimization method further, trials were performed on a benchmark truss problem domain and the results obtained were compared with results obtained by other researchers.

The results of the performance evaluation trials for the proposed method, in which the sizing, shape and topology were simultaneously performed, indicated that the method was effective in evolving a variety of truss topologies compared to previous published results, which evolved from a ground structure. The diverse topologies, however, were obtained over several trials instead of being found in a Pareto-optimal set found by a single trial. In addition, the proposed method was not able to locally optimize the member section sizes. Additional trials were performed to determine the benefit of applying local optimization to the member section sizes for a given truss topology or geometry provided by the method. The results indicate that significant weight reduction could be achieved by performing local optimization to the truss designs obtained by the proposed multi-objective optimization method.

ACKNOWLEDGEMENTS

I would like to thank Dr. Anne M. Raich for her guidance and support throughout my graduate studies and research and the enormous effort she made to revise my entire document. I am especially grateful to her for the freedom she gave to explore this topic while helping me to maintain focus and overcome difficulties.

I also extend my heartiest gratitude to Dr. Bracci and Dr. Anshelevich and Dr. Strouboulis for serving on my committee and giving me helpful suggestions to improve my document.

Additionally, I wish to acknowledge Mr. Sangwook Paik, Mr. Pranab Agarwal, Mr. Sachin Mattakari and Dr. Breanna Bailey for their help and support throughout the research.

Finally, I would like to thank my parents, my sister, Ms. Pragyansri Pathi, my brother-in-law, Mr. Sasi Kuppannagari, my best friend Suhani Shah and all my friends for their love and encouragement all the time.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

CHAPTER I
INTRODUCTION

Overview

Performing engineering design optimization, especially the optimization of complex structural systems, is an active area of research. The focus of research in structures is on developing computational methods to support decision making concerning complicated modern structures that carry heavy loads over long spans, while also meeting specific architectural, construction, and economic requirements. As part of many of the methods developed, there is a need to support the exploration of a broad range of conceptual designs before making design decisions. Therefore, the methods developed must be able to realistically define numerous design variables, support techniques to locate global or near-global optimal designs, and improve the efficiency of the computational procedures implemented. This research effort focuses on the design optimization of long span truss systems. The computational methods developed seek to obtain cost-efficient designs, with respect to construction and material costs, while also satisfying structural design requirements and a designer's aesthetic criteria concerning layout of members and joints. Consideration of all of these issues makes the development of design and optimization methods a challenge even for smaller-scale structural systems.

Modern researchers have been using Genetic Algorithm (GAs) (Holland 1975; Goldberg 1989) and other heuristic methods to search for near-optimal topology layouts, in addition to determining member cross-sectional areas. Lin and Hajela (1992, 1993) performed discrete shape optimization to find the minimum weight of an eight bar truss subjected to displacement constraints. Rajeev and Krishnamoorthy (1992) used GAs to find minimum weight-truss systems through discrete shape optimization subject to stress constraints. Sakamoto and Oda (1993) used GAs to find minimum weight trusses by optimizing truss topology by combining genetic algorithms with an optimality criteria

This thesis follows the style and format of the *Journal of Structural Engineering.*

method. GAs were used to perform the layout of the truss and the optimality criteria assisted in finding the member cross-sectional areas. In comparison, Hajela and Lin (1994) adapted a two-stage shape and topology optimization process. The first process found the optimal topology of the trusses using a ground structure approach, and the second process found the optimum member sizes for the resulting truss topology. Many other heuristic optimization methods have been employed by researchers in the field of truss shape and topology optimization. One of the most common example includes fuzzy logic-controlled genetic programming (Yang & Soh, 2000).

Even though there has been a substantial amount of research performed in the field of truss shape and topology optimization, research that seeks to simultaneously optimize the truss geometry, shape, and topology is uncommon. In addition, many research efforts remain focused on problems that work with a predefined structured domain, in which the number of nodes and members and the load locations are fixed. In order to fully explore conceptual designs for a problem, however, designs that vary in the number of members and nodes and in how the loads are carried by the structure must be examined, which requires that the limits imposed by a predefined structured domain must be removed.

Motivation

This research effort is motivated by the need to consider structural designs that may be beyond the established conventions of designers in the search for cost-efficient, structurally-sound designs. Mathematical optimization methods work with inflexible statements of constraints in the formulation and perform a local search, which may result in sub-optimal design alternatives. Therefore, these methods are not suitable for generating varied design geometries and topologies and performing a global search for near-optimal design alternatives. A computational design program, however, can be developed to provide the ideal environment for exploring more efficient and varied designs. Genetic algorithms, along with other heuristic methods like simulated annealing and taboo search, have been shown to provide global exploration capabilities and have been used for design optimization. Many researchers have worked on problems that are limited to structured domain, in which the all the trusses optimized have the same fixed nodal locations, number of members, number of nodes, and load locations. Some researchers have also worked on the optimization of all the three levels of optimization (i.e. topology, geometry and shape), but have only done so in stages. Research focused on developing methods to simultaneously optimize all three levels of optimization is only just beginning to appear in the context of structural design problems. All of these factors form the motivation to develop a computational method that can efficiently search over a broad range of different truss geometries and topologies while at the same time performing all three levels of truss optimization simultaneously.

Research objective and scope

The objective of this research is to develop a flexible design grammar and genetic algorithm representation to be used by a computational method to design efficient steel roof trusses given design space dimensions and loading requirements by the user. The goal of the computational method is to obtain a set of near-optimal trusses for the defined unstructured problem domain, not just a single near-optimal design. The unstructured domain only prescribes the magnitude of loading, the support locations, and the overall dimensions of the domain. No other structural information concerning nodal locations or the number or the placement of members is defined.

The method developed will support the optimization of truss designs for two objectives that concern weight, and deflection along with keeping the constraints of stress within the prescribed limit. Multi-objective genetic algorithms are used in this research effort to evolve the set of Pareto-optimal trusses within this domain that have varying topology, geometry, and sizes. A Pareto-optimal curve (weight and deflection), which represents the optimal set of design alternatives evolved, will be generated for each predefined unstructured problem domain (loading condition, support location, and maximum height) investigated in this research. The Pareto-optimal curve will be used to provide design engineers with a range of near-optimal design alternatives that show the tradeoffs that occur in meeting the stated objectives. Designers can select their final design from this set based on their own individual weighting of the design objectives. The quality of the designs obtained will be evaluated by comparing the truss designs evolved using the proposed computational design system with truss designs obtained by other researchers on a defined benchmark problem domain cited in the research literature.

Methodology

The computational search method and design representation developed in this research is based on Multi-objective Genetic Algorithms (MOGA) (Goldberg 1989). In single-objective optimization, typically one near-optimal solution is desired and this solution is the best at meeting the stated objective. However, most realistic engineering problems have multiple, often conflicting, design objectives. If there are several design objectives, a single criterion can be formed in a GA by using one composite fitness function that is created by weighting each objective. The best single design alternative can then be found by optimizing the composite fitness function. To support the search for an efficient design starting back at the conceptual design phase, however, it is not critical, and probably not even desired, to search for only a single solution. Instead, it is more important to provide design engineers with a view of the broad range of design alternative designs that result from different design objective priorities. MOGAs can be used to evolve a Pareto-optimal set of designs that optimize several objectives, which enables the engineers to evaluate the tradeoffs that occur due to having conflicting objectives.

To obtain the Pareto-optimal set of truss designs that minimize weight and deflection while maximizing member stress ratios, several multi-objective concepts proposed by other researchers are applied in this research. The search domain of the design problem studied has many multiple local optimums. Even small changes in the topology, geometry or member sizes can often create a locally-optimum design. In this large and complex search domain, there is a high possibility that many locally optimum trusses are prematurely eliminated from the population even though they may eventually have better structural performance than others when further optimized. Several mechanisms must be used, including fitness sharing, to prevent the population from converging to a single near-optimal truss design and to maintain solutions that are well distributed all over the entire Pareto-front during the entire GA process. By providing these features, a diverse range of topologies and shapes of trusses will be evolved.

Outline

Chapter I provides a discussion of the motivation, objective, scope and methodology for this research effort. Chapter II provides a literature review of work done by previous researchers on structural design optimization using GAs and MOGAs , in addition to discussing the basic operators of a GA. In Chapter III, the unstructured design problem domain investigated in this research is defined and the MOGA representation used for truss optimization in this research is presented. Chapter IV details the procedures involved in implementing the developed methodology and the experiments performed to verify that the method is capable of obtaining efficient truss design alternatives. Chapter V defines the benchmark problem investigated and provides a comparison of results obtained from the computational method developed in this research with the research obtained previously by other researchers. The conclusion is presented in Chapter VI, which provides a summary of the results, including a discussion of the benefits provided by the proposed method, the problems identified by this research, and future recommendations to extend this research work.

CHAPTER II
LITERATURE REVIEW

This chapter is divided into three parts. In the first part, the layout optimization of truss system is discussed briefly. The second part highlights on the format and operators used for simple genetic algorithm (SGA) and the multi-objective genetic algorithm (MOGA). In the third part, a review of the methods and strategies proposed by other researchers in solving truss optimization problems using GAs is presented.

Layout optimization

Layout Optimization of truss systems refers to producing trusses with minimal weight and primarily satisfying stress, displacement, and slenderness criteria. Layout optimization can be classified into three main categories:

I) Sizing Optimization: Sizing optimization is concerned with the optimal selection of the cross–sectional areas of the truss members where all the nodal connectivity and the nodal locations remain fixed in the truss.

II) Geometry Optimization: Geometry optimization is concerned with the optimal selection of the location of the nodal connections, which affect the lengths of individual members and the overall shape of the truss, which primarily refers to changing the nodal co-ordinates during optimization.

III) Topology Optimization: Topology optimization is concerned primarily with the placement of members in the truss structure relative to each other. It also determines the number of members and the number of nodes that exist in the structure along with their support conditions.

Overview of genetic algorithms

Introduction

The solutions obtained from mathematical search methods studied substantially in the past are based on a local scope and the final optimal solution found depends heavily on the initial starting point and on the neighborhood of the search space investigated.

However, most real world problems have unpredictable, complex domains, in which continuity and existence of mathematical function gradients are not guaranteed. For these reasons, the mathematical methods developed for optimization do not have robustness required to search structural design domains that tend to have non-convex, highly nonlinear search spaces with small, often discontinuous, feasible regions.

Genetic algorithms (GA) were first developed by Holland (1975) and his colleagues and his students at the University of Michigan. They were further developed by Goldberg (1989). GAs works on the principle of natural selection to evolve solutions to problems. The GA operations can be divided into those that operate in the Genospace where the genetic operations are performed and those that operate in the Phenospace, where the function evaluations are performed on individuals. The GA fitness function serves as a link between the two realms, since it determines which encoded individuals have a better chance of survival based on their respective objective values. The other link between the two is the encoding scheme that uniquely maps an individual design solution from the phenospace to its corresponding bit-encoded genotype string and vice versa (Goldberg 1989). Thus these systems have the characteristics that they evolve through the application of genetic operators of recombination and mutation and adapt themselves to the environment through fitness selection. Many researchers have validated the usefulness of GA's in solving optimization problems. The best part of applying GAs is that they are easy to implement computationally, while at the same time providing powerful search methods. Goldberg (1989) identified four main characteristics in which GA are different from other traditional optimization methods:

1) GAs work with a coding of the parameters set, not the parameters themselves
2) GAs search from a population of points, not a single point
3) GAs use payoff (objective function) information, not derivatives or other auxiliary knowledge.
4) GAs use probabilistic transition rules, not deterministic rules

Simple Genetic Algorithm (SGA)

Fig. 2.1 below presents the flowchart of the operators involved in implementing the SGA. The actions of the operators as shown allow the SGA to search for improved solutions over the generations performed. Individual design solutions are typically represented in the genotype by a bit encoding of 0's and 1's in a string, although other types of encodings are possible. The evolutionary process starts from a population of randomly generation individuals and proceeds in a generational fashion. In each generation, the fitness of the whole population is evaluated and each individual is assigned fitness values depending on how well that individual meets the stated objectives and constraints. Selection is then performed to pick individuals from the current population based on their fitness values to form a new population. This population becomes the current population in the next generation.

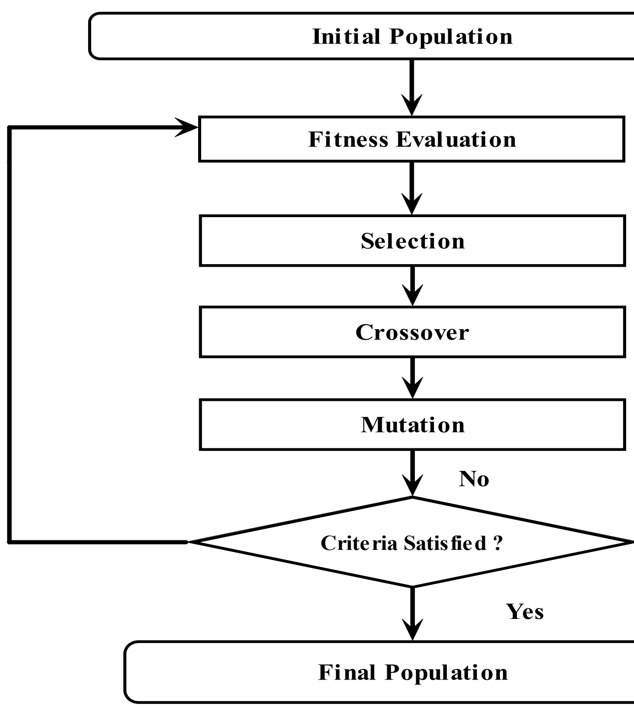


Fig. 2.1. Flowchart of simple genetic algorithm

SGAs apply three main genetic operations: selection, crossover, and mutation. De Jong (1975) conducted research to determine the performance of GAs that implemented simple crossover, simple mutation, and roulette wheel selection using five test functions. The form of the GA studied from the basis of the SGA. Since then SGAs have been used in a wide variety of applications with different combinations of SGA operators, parameter settings and search strategies. The operations shown in Fig. 2.1 are repeatedly applied until no further improvement is found or a pre-specified number of generations is performed.

Simple Genetic Algorithm (SGA) representation

All optimization design variables for a single solution are represented by binary numbers in a SGA individual. Fig. 2.2 presents two design variables (X1 and X2) that are converted into binary numbers, in which each variable is encoded using 3-bits. The value range of each variable in this case is 0 – (2^3 -1). An individual is then defined by concatenating these binary numbers together to form a string.

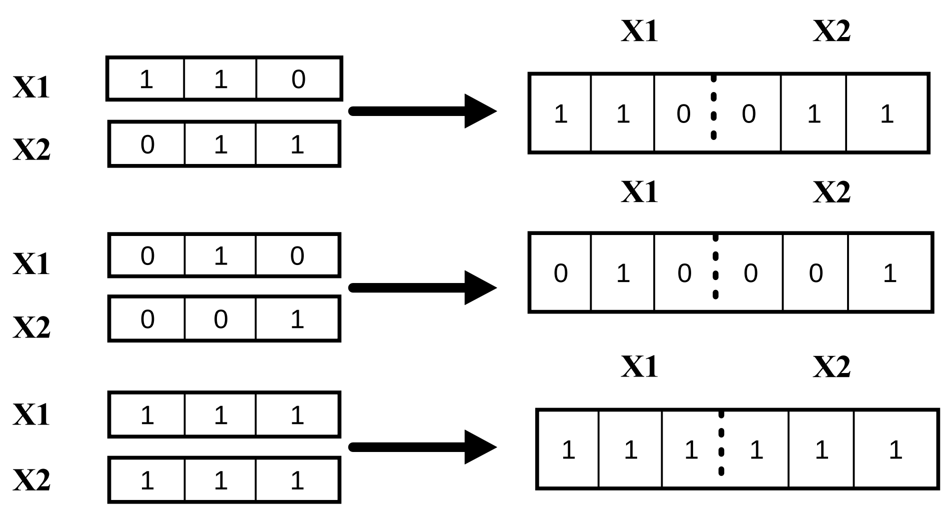


Fig. 2.2. Representation of SGA

Fitness evaluation

Each individual in the population is randomly initialized. These individuals are decoded during the process of optimization and the decoded values are used to evaluate the fitness of each individual. The fitness of each individual is evaluated using a fitness function. The fitness function performs the function of determining how well the individuals meet the stated objectives along with the penalties on fitness for violation of constraints. For example, SGA can be applied to an optimization problem with a single objective function:

$$\text{Max } F(x) = \sum_{i=1}^{n} X_i \tag{2.1}$$

The fitness value of each individual in the SGA population is then computed using equation (2.1). Table 2.1 shows the fitness values of each individual.

Table 2.1. Fitness values of individuals computed with equation 2.2 and values in Fig. 2.2.

Individual	X1	X2	Fitness (X1+X2)
1	6	3	9
2	2	1	3
3	7	7	14

Selection operator

Based on their assigned fitness values, some individuals in the current population are selected to be part of the population evaluated in the next generation. Individuals with higher fitness values have a greater chance of being selected than those having lower fitness values. The selected individuals are used to generate new offspring through genetic operators (crossover and mutation) to construct the population for the next generation. Roulette wheel and tournament selection are the two important selection schemes that are most common in the literature. In this research, tournament selection is used to select individuals to undergo crossover and mutation. In general, in tournament selection 'N' (where 'N' is a predefined number) number of individuals is selected at random from among the current population and the fittest among them is picked for the new population. The process of tournament selection is continued until the next generation population is filled. Fig. 2.3 provides a visual example of tournament selection where N = 3. Individuals having higher fitness values have a greater chance of being selected, thus resulting in generating fitter individuals in the next generation. Tournament selection allows user to have greater control over the selection pressure from one generation to the next. A smaller tournament selection will have less selection pressure where as a higher tournament selection will have high selection pressure which typically results in obtaining local optima.

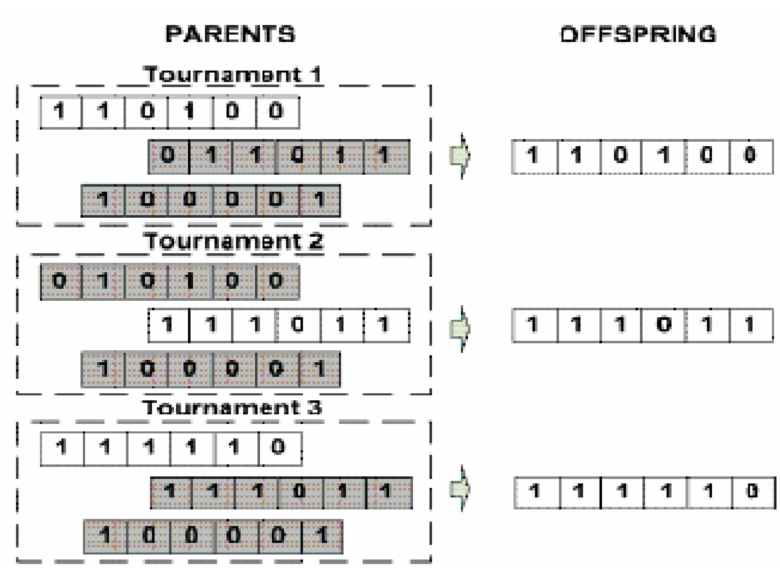


Fig. 2.3. Example of tournament selection (tournament size: 3)

Genetic crossover and mutation operators

The selected individuals in the population undergo possible alteration using crossover and mutation operators. Many crossover techniques exist, including single point crossover, multipoint crossover, and uniform crossover (Goldberg 1989). Crossover is a recombination operator. It cannot produce new information to the search process, but it can provide a mechanism for individuals to improve their fitness. For example using single point crossover, a point on the parent individual is selected and all encoded bits beyond that point in the individual string are swapped between the two parent individuals. The resulting individuals are called children and wait to undergo possible mutation before the next generation. Fig. 2.4 presents an example of single point crossover applied to two individuals.

Chromosome 1	11011	001001111
Chromosome 2	00111	110011110
Offspring 1	11011	110011110
Offspring 2	00111	001001111

Fig. 2.4. Example of single point crossover

Mutation is used in GAs to help maintain genetic diversity from one generation to the next, and also to help introduce or reintroduce useful information into the population of individuals. The purpose of mutation in SGAs is to allow algorithms to avoid local minima by preventing the population of chromosomes from becoming similar to each other, thus slowing or even stopping evolution. Fig. 2.5 presents a simple mutation operation, which involves flipping each bit in the encoding based on a set probability of mutation.

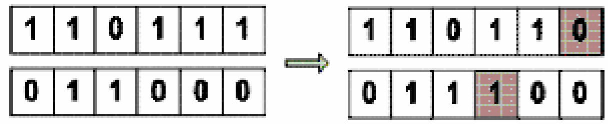


Fig. 2.5. Mutation operation

Elitism

The SGA operators and selection mechanisms are all applied probabilistically. Therefore, there is no guarantee that the best individuals in the current populations will be selected. Elitism is used to transfer copies of the best individuals in the population directly into the next generation without undergoing any crossover or mutation. The rest of the population is selected using tournament selection. Using elitism prevents the loss of the best solutions found to date by the SGA.

Multi objective genetic algorithm (MOGA)

In SGAs, a composite fitness function must be used to handle problems in which there is more than one stated design objective. The individual objectives are combined together through addition or multiplication and often exponential or other factors are applied to impose the desired priority among the satisfaction of the different objectives. The composite fitness function plays an important role in determining whether an individual is selected for reproduction to produce the individuals in the next generation. However, in the composite function often becomes highly sensitive to its formulation and to the factors applied. Therefore, often a lot of effort must be put into optimizing the composite function form to obtain good, consistent results. In addition, the composite fitness function with a predefined set of factors can only be used to obtain a single solution. Therefore, the major advantage of using a MOGA over a SGA is its independence of the fitness function to the priority factors used to handle multiple objectives and the ability to obtain a set of equally good solutions. In MOGA, the selection of parents to undergo reproduction is done on the basis of their individual rank, which in turn is determined from their distance to and position on the trade-off surface obtained considering the stated objectives separately. This surface, commonly called the Pareto surface, is a three dimensional curve in this research since there are three objectives i.e. stress, deflection and weight. Those individuals in the population that are determined to be non-dominated by all other individuals are assigned Rank 1. During ranking, these individuals are then eliminated from the ranking process (having already been assigned a rank of 1) and the next set of non-dominated individuals in the population is found and assigned Rank 2. The process continues till all the individuals in the population are ranked (Srinivas and Deb 1994).

Overview of previous research on truss optimization using genetic algorithms

Traditional and evolutionary approach to truss optimization

Traditional methods of optimization were the main focus of research during the late seventies and eighties. This research led to the application of a number of mathematical optimization techniques to optimize trusses. Most of the research performed concentrated on member size optimization for a fixed structural topology and geometry. Berke and Khot (1987) used optimality criteria method and Schmit (1981) used a mathematical programming method for truss optimization. Templeman and Yates (1983) suggested a method for discrete optimization using segmental members. John and Ramakrishnan (1987) studied a combinatorial optimization approach using branch and bound algorithms for discrete structural truss optimization.

Rechenberg (1965) was the first to use the concept of biological evolution in design and analysis, although the importance of GAs came into picture through the work of Holland (1975). Goldberg (1989) researched the applicability of genetic algorithm in the optimization field which started an era of research in this field.

Shape, geometry and topology optimization

Rajeev and Krishnamoorthy (1992) and Jenkins (1992) were among the first to apply GAs successfully to truss optimization. The former researchers concentrated their efforts on the optimization of member sizes; whereas the later researcher focused on geometry optimization. After the successful implementation of GA in shape optimization of trusses, many researchers tried to simultaneously apply topology and geometry optimization along with the shape optimization in order to enhance the ability of design optimization.

Several researchers in the mid-1990s tried to work on the limited topology and geometry optimization from different prospective. Hajela and Lee (1994) performed topology optimization using the ground structure approach, which was first proposed by Dorn et al. In their research, they followed a two stage procedure. In the first stage a number of truss topologies that were kinematically stable were generated. The next stage used the topologies generated in the first stage as initial seeds and member-sizing optimization was performed with the consideration of structural constraints in addition to additional topology optimization. In order to overcome the two stage procedure, Rajan (1995) using the ground structure performed sizing optimization along with limited geometry and topology optimization. Although the research was an improvement over the previous ones concerning optimization of truss designs, it still had the drawback of using a ground structure approach, which limited the flexibility of creating diverse designs in an unstructured problem domain.

Other researchers also tried to implement fitness sharing as an effective technique to maintain the diversity in the population and curtail the need of large population sizes and longer string lengths. Gage et al. (1995) developed a variable- complexity genetic algorithm procedure for topology optimization. Their method using GAs was followed up by a gradient based optimizer to size the members. This research effort also introduced the variable length GA and the use of cut-splice crossover operation. Further improvement in the field was shown by Deb and Gulati (2001), in which each member's presence or absence in the truss topology was determined by the area assigned to each member of a truss. If the member size was less than the predefined critical area, then the member was removed. In their research, the nodes of the trusses were divided into two categories. The first category was known as the basic nodes and the next as non-basic nodes. The presence of the basic nodes was a must to make the trusses kinematically stable. The non-basic nodes were optional. Thus, the trusses that did not have the basic nodes were excluded from further evaluation.

Optimization in an unstructured problem domain

In the research conducted by Shresta and Ghaboussi (1998), a new methodology to evolve optimum truss design in an unstructured problem domain, in which the designs are allowed to emerge free of preconceived designs, was developed. The string representing the structure is made up of fixed number of concatenated substrings. Each substring encoded the nodal location, the presence or absence of a node and the member information for members that are connected to the node. Even though a fixed length string was used, the maximum number of nodes was kept as a variable as the string length could be initialized to the desired extent. To evaluate the efficiency of the proposed methodology, optimization trials were performed for two design spaces that had maximum heights of 10m and 35m with both spanning 70m in length. Weight optimization of the truss was performed taking into consideration the different types of design constraints. The trusses generated from the two design spaces had reasonable configurations and the stresses in the members were within the specified limits set by the prescribed loading and design space. Thus, this research was a milestone in the field of adaptability of SGA to unstructured problem domains and to truss topology and geometry optimization.

Hybrid heuristic techniques

During the late nineties various researchers have come up with a wide variety of ideas and ways of implementing GAs to improve the optimization of trusses. Rajeev and Krishnamoorthy (1997) proposed a strategy that resulted in automatically arriving at lower bound indices for each design variable along with using a variable length genetic algorithm. The topologies that did not result in optimal solutions died off in early generations, and after a few generations the population was found to be composed of individuals of the same lengths.

String lengths play a major role in determining the computational time involved in the optimization process. Long string lengths make the GA process slower in general. This is a serious problem when large scale optimization problems are considered. Jenkins (1997) proposed in his research a space condensation heuristic that helped in achieving better optimal results in a shorter period of time. Jenkins (2002) developed an idea of implementing GA without crossover. He proposed an adaptive GA that used only mutation. In his research, two kinds of mutation were used. One was the random mutation, which was similar to that used in SGAs, in which mutation is performed with genes that are selected based on mutation probability, and the other was intelligent mutation that was performed conditionally. For instance, if the stress in a truss member exceeds the design stress then positive mutation was applied to increase the cross sectional area otherwise negative mutation is performed. Thus, this type of implementation reduced the computational expense of the method.

Yeh (1999) proposed a hybrid GA to enhance the efficiency of the GA process. The concept of fully stressed design is combined with that of the survival of the fittest. Fully stressed design is efficient in finding local optimum, while GAs are better at finding global optimum. This research achieved faster convergence and more stability as compared to the use of GA alone.

Raich and Ghaboussi (2000) used an implicit redundant representation (IRR) of the GA string to produce near-optimal frame designs. The IRR GA overcomes the fixed parameter limitation of SGA by its ability to self organize the GA representation and to enable the individuals in the population to encode a varying number of design parameters during optimization. This methodology was able to expand the optimization search process to include diverse topologies and geometries simultaneously.

CHAPTER III
DESIGNING A PARAMETER VALUE REPRESENTATION AND MODELING AN UNSTRUCTURED DOMAIN

The design representation used by a GA is critical in that it provides the mapping between the genotype string encoding and the phenotype expression of the design variable values. In order to generate a truss structure and to analyze it, all the information about the design parameters like the nodal locations, number of nodes, and member areas must be encoded in an individual. In addition to the information regarding the parameters, loading information, including the magnitude and location, and the support conditions must be specified to completely analyze the generated truss structure. Therefore, for modeling trusses in an unstructured domain, the GA population must provide all the information required to define a diverse set of complete truss structures to evaluate each generation.

To define each truss alternative, each individual's information is encoded as a binary string that can be decoded into the design variable values. A predefined mapping between the binary representation and the design variables, which is called the design grammar, is defined. A design grammar plays an important role in the whole process of design and optimization. Its definition strongly influences the shape and topology of the trusses generated, which implies it has a strong influence on the size of the search space. It also determines the proportion of the feasible area to that of the infeasible area in the search domain. Taking the above facts into consideration implies that the choice of the design grammar is one of the most important aspects of developing a computational method to assist in optimizing truss designs.

Many researchers have worked on various types of design representations in attempts to provide a flexible encoding of design variables. As discussed in Chapter II, a ground structure approach uses a predefined topology and geometry for fixed numbers and locations of nodes. Each member's presence/absence is determined by an 'on/off' bit (1/0) in the GA individual. Limited geometry optimization has also been performed (Rajan 1995) by considering the location of some nodes as design variables. Robert et al.

(1996) in his research used a triangular element that represented sub-geometry of a simple bridge element. A number of triangles were joined at their baseline and the top nodes of each triangle were connected to generate a stable structure.

In addition to the design representation it is necessary to define the problem domain. The individuals (truss alternatives) in the GA population need to compete with each other based on their fitness in their environment, which very much depends on the problem domain. For structural problems, there must be boundary conditions applied to limit the search space as otherwise it would make the search space infinite. In this research, the maximum height of the truss and the maximum span length are defined to help constrain the search space.

The main objective of this research is to investigate the performance of a new design representation by evaluating the impact the design representation has on obtaining quality sets of Pareto-optimal trusses using MOGA. A flexible design representation has been developed to achieve this goal. The unstructured domain is constrained using boundary conditions to limit the search space and to ensure designs that have practical application.

Designing a parameter value representation

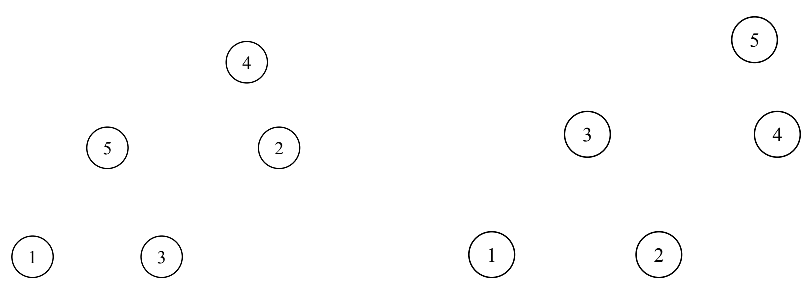
Several researchers have proposed different forms of design variable representations in their research. This research implements a form of the implicit redundant representation (IRR) GA (Raich & Ghaboussi, 1997). In the IRR GA, a single individual is composed of a number of redundant segments and gene instances, which contain the encoded design variable values. The gene instances contain essential information, which is used to generate a truss alternative based on the decoding process defined by the design grammar. The location of gene instances is explicitly specified in the individual. The location and contents of the redundant segments do not affect the gene instances. Fig. 3.1 represents the design variable representation used in this research.

Max. No. of Nodes	Nodal Locations	Member sizes	Redundant Segments

Fig. 3.1. Parameter value representation used in this research

The first four bits of the string encode a number that defines half the maximum number of nodes to be used in the truss formation. The next nine segments encode the location of each node (which assigns the 'x' and 'y' co-ordinates). Then after the nodal locations are assigned and the truss is generated using the predefined design grammar, the member sizes are assigned to the members using the next set of string segments.

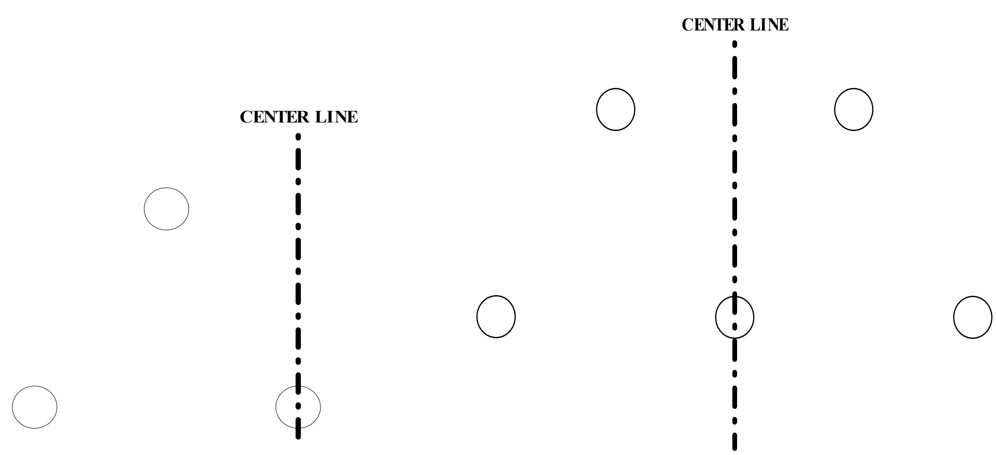
In this research, a maximum value of the string length is assigned taking into consideration the maximum value of nodes that can be generated and the maximum number of elements that the truss can have for the maximum number of nodes generated. The trusses are generated based on the nodal locations. After the nodes are generated they are sorted according to their distance from the base node, which in this research was fixed at the origin (x coordinate and y coordinate both at 0). Then the nodes are numbered starting from the base coordinate as the first node according to their distance from the previous node. Fig. 3.2 shows how the node are decoded, sorted and numbered with respect to their previous nodes, keeping the base node at the origin.



(1) Nodes generated by GA **(2) Nodes numbered based on distance**

Fig. 3.2. Sorting and numbering of nodes based on their distance

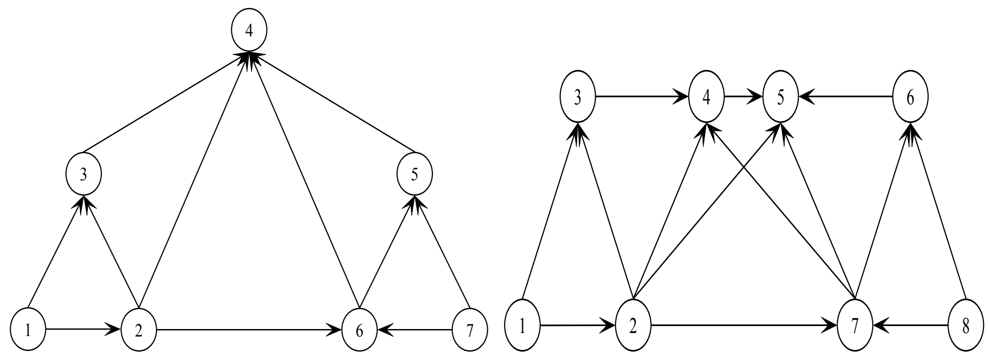
This research only investigates symmetric truss structures. Constraints that require symmetry of members and nodal locations could be defined. Instead, in this research, the representation only generates one half of the nodes. By mirroring the generated information to the other side, a complete set of nodes required to evolve the whole truss structure is generated. The reduced problem domain resulting from using the imposed symmetry saves computational expense. The degrees of freedom associated with each node are assigned to the nodes. Fig. 3.3 shows the generation of half of the nodes and the imposed symmetry to create a complete truss structure.



(1) Nodes created (2) Impose symmetry

Fig 3.3. Nodes generation using the nodal information decoded from the MOGA genotype

The generated nodes are connected to each other by a predefined design grammar to develop a truss structure. Various types of design grammars were investigated initially in this research, but the final design grammar selected was the one that satisfied the desired criteria (overall flexibility in changing the number of nodes and members, overall desired characteristics in topology and geometry shown in generated trusses).



(1) Nodal Connectivity with Centre node **(2) Nodal Connectivity without Centre node**

Fig 3.4. Design grammar used in this research

Fig. 3.4 shows the schematic representation of the design grammar used in this research. The nodes are connected to each other by preference of their numbering through a predefined pattern. The 1st node is connected to the 2nd and the 1st node is also connected to the 3rd. Similarly the 2nd node is connected to the 3rd and the 4th node. The process continues until the first half of the truss is completely connected. The other half of the truss (the mirror image) is connected in a similar manner. In order to connect the two sides of the truss, a set of criteria are applied to determine the type of connection to impose.

- 1st criteria: If there is no node at the midpoint: Two nodes are selected such that one is nearest to the midpoint at the bottommost level and the other is closest to the midpoint at its topmost point. Then the nodes corresponding to these nodes on the other half is selected. These nodes are all connected each other in a cross pattern.
- 2nd criteria: If there is one node at the center line: First two nodes are selected excluding the one at the center line, such that one is nearest to the midpoint at the bottommost level and the other is closest to the midpoint at its topmost level. Then the nodes are compared with respect to their distance with respect to the

centerline node. The one which is closest to the center line node is dropped and the other one is taken into consideration for connecting the two halves by finding a corresponding node on the other half.

After the nodes are connected together, the members are assigned areas from a set of predefined set of member areas, such that the assigned values of member properties are symmetric. At the end of this process, a symmetric truss structure is defined.

Since the number of loaded nodes and their locations is not fixed, the loads are specified by the user as distributed loads instead of point loads. For each truss examined by the MOGA, the top nodes in the truss are determined and each of the top nodes is assigned a load based on the tributary area method. A distributed load of 3 k/ft is applied to all the trusses examined in the trials presented, which assumes a spacing of 10 ft between planar trusses. Thereafter, the truss is checked for the stability criteria and the satisfaction of dimension constraints. The trusses which pass are sent on for analysis, wherein the stresses in each member and the overall deflection of the truss is calculated for the given support conditions. Fig. 3.5 presents how the tributary area method is used to determine point loads on the top nodes from the distributed load.

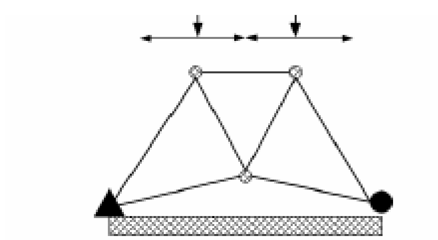


Fig. 3.5. Tributary area method used to determine the loading on the top nodes

Modeling in an unstructured domain

An unstructured design domain has no prescribed bounds on the number of design parameters as compared to structured design domain. The unstructured design domain reflects a realistic design domain that has no a priori knowledge concerning the number of design parameters. In comparison, a structured design domain fixes the values for many design parameters and cannot change these values during optimization. Specifically for truss design, the unstructured design domain has no predefined information regarding the nodal locations, the number of nodes, the member sizes neither it has any knowledge of the load location nor the magnitude of the loads on the loaded nodes. In the unstructured problem domain, the number of nodes and their locations are allowed to vary. Therefore, the load is assigned as distributed load instead of point load on specified nodes. When the design parameters are decoded from MOGA individuals to obtain the nodal locations then the distributed load is converted into point loads on each top node based on the tributary area. Only minimal design information is specified for the unstructured domain in order to impose the designer's criteria on dimensions and support types. In this research, the minimal design information provided is limited to:

- Maximum height of the truss : 15 ft
- Span length: 40 ft and 60 ft
- Support Conditions: Hinge and Roller
- Load: 3 k/ft

Fig. 3.6 presents the unstructured design domain and imposed load conditions used in this research. The experimental trials performed for the benchmark problem use a different problem domain definition. The criteria and design information for the benchmark problem is presented in Chapter V.

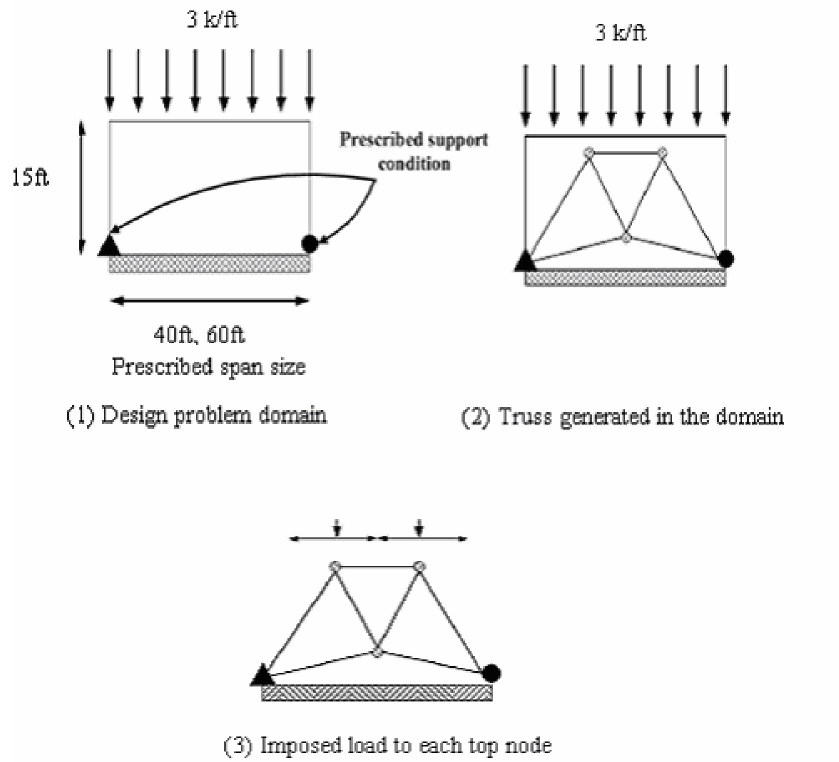


Fig. 3.6. Unstructured design domain and imposed load conditions

CHAPTER IV

MULTI-OBJECTIVE OPTIMIZATION USING NON-DOMINATED PARETO OPTIMAL METHODS

Background information

Multi-objective optimization attained by using a composite fitness function that is an aggregation of multiple objective values results in convergence of the GA population to a single solution. As the number of objectives increases, the optimization of the single solution becomes difficult due to tradeoffs that exist among the objectives. In addition, the single solution obtained only reflects one possible weighting of the objectives. Therefore, when a constant weight is assigned to each objective function, which is required to construct the composite fitness function, the direction of the GA search is fixed in the multi-dimensional space as shown in Fig. 4.1. In Fig. 4.1, f^1 is an objective function to be maximized and f^2 is to be minimized. The close circle in Fig. 1 represents the final, single solution obtained by the GA. In design, the relative weighting of the objectives is impossible to do a priori. Therefore, obtaining a single solution will not necessarily best satisfy the designer's priorities in satisfying the multiple objectives. To address this problem, each objective should be considered as equally important as opposed to combining all the objectives into one measure through weighting. This approach enables the discovery of an optimal set composed of solutions that are equally good at optimizing conflicting objectives specified. The optimal set will contain more than one solution. These solutions will define a range of designs that are obtained in a single GA trial, instead of many individual GA trials with different defined weightings of the objectives.

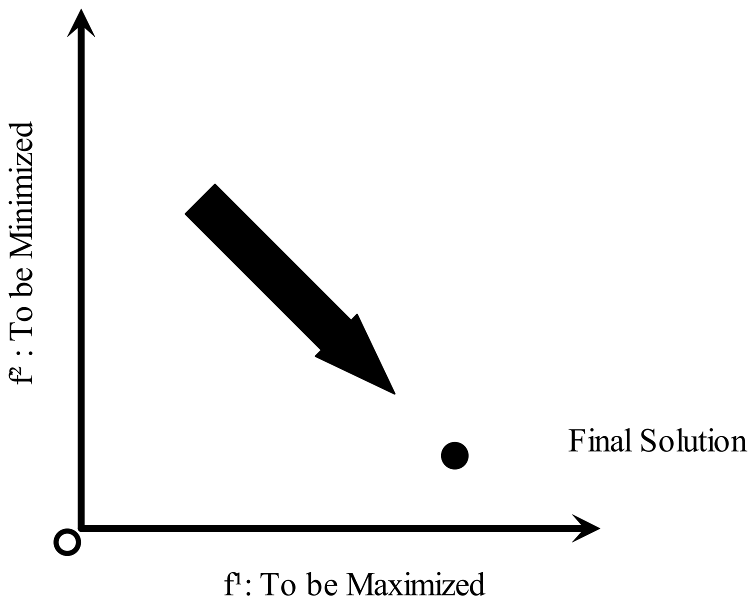


Fig. 4.1. Direction of search in GA with a combined fitness function

Schaffer (1985) made an early attempt to perform optimization using multi-objective genetic algorithms (MOGA). Schaffer proposed the vector evaluated genetic algorithm (VEGA) for finding Pareto-optimal solutions for multi-objective optimization problems. In VEGA, a population is divided into disjoint subpopulations that are governed by different objective functions. Although Schaffer reported some successful results, VEGA can only find extreme solutions on the Pareto front since the search direction is parallel to the axes of the objective space. Fig. 4.2 shows the search direction of VEGA. To improve on the above shortcoming, Schaffer suggested two approaches. One is to provide a heuristic selection preference for non-dominated individuals in each generation and the other is to crossbreed among the species by adding some mate selection.

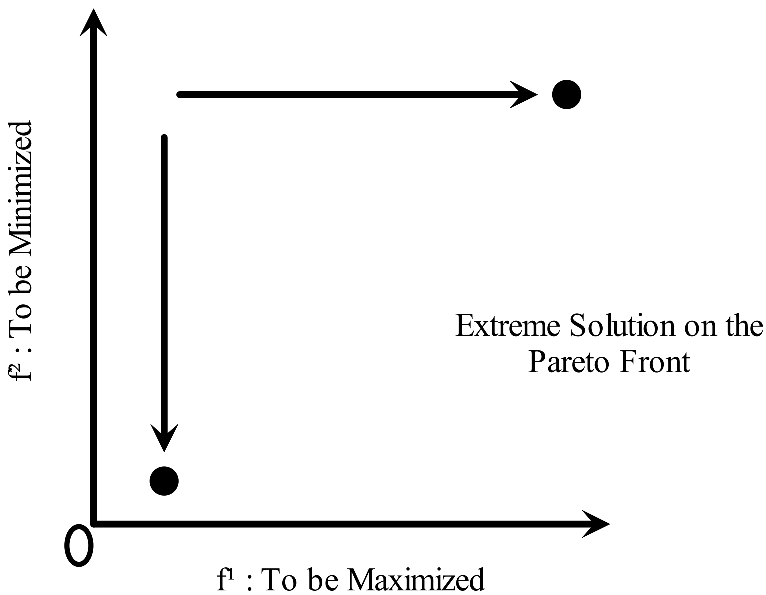


Fig. 4.2. Direction of search in VEGA

Goldberg (1989) suggested a novel non-dominated sorting procedure, which uses the concept of domination to give preference to non-dominated individuals in the population. Goldberg also suggested the use of a niching strategy among solutions of a non-dominated class to satisfy the second goal of multi-objective optimization i.e. diversity among solutions. The niche count (niching strategy) is a measure of how similar an individual is to other individuals in the population. Imposing a selection biased to the niche count enables GA individuals in a population to be better distributed all over Pareto front by providing the information of how much an individual is shared in a population rather than having clumps of individuals that are similar form along the front. In this research, niche count was implemented by using sharing functions (Goldberg 1989).

Non-Dominated pareto ranking and sharing functions

Goldberg (1989) suggested non-dominated Pareto ranking, in which all the non-dominated individuals in the population are assigned the same fitness for selection. Non-dominated individuals are those that are better than or equal in objective fitness to all other individuals in the population. By comparing all the individuals with one another, the highest rank can be assigned to the non-dominated individuals. Excluding the highest rank individuals, the same procedure is executed to assign the next highest rank to the non-dominated individuals remaining in the population. The procedure is repeated until all the individuals are ranked. Non-dominated Pareto ranking provides a rational way to consider all the conflicting objectives that cannot be compared directly with each other. The concept of non-domination is illustrated in Fig. 4.3, which shows a set of six sample solutions for two criteria (f^1 and f^2). The non-dominated solutions in the set are indicated by a ranking of zero (0). For each of the non-dominated solutions, there is no other solution in the set that has a lower value in both criteria for this minimizing problem. However, the solution ranked four (4) is dominated by four other solutions in the set, i.e. four other solutions have lower values in both criteria.

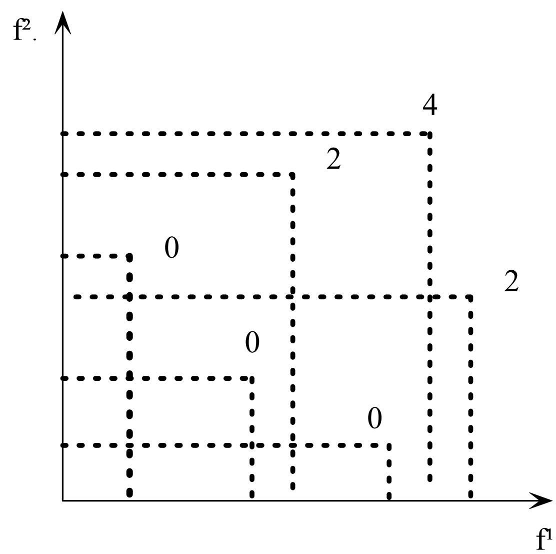


Fig. 4.3. A Pareto ranking scheme

To better support the distribution of individuals over the entire Pareto-optimal surface, a sharing function can be applied. The sharing function defines the degree of fitness sharing that must be applied to each individual in a population (Goldberg 1989). Highly similar individuals in the population are penalized by a reduction in fitness. Reducing an individual's fitness increases the population diversity pressure of selection for the population, which allows the population to maintain individuals at local optima for multi-modal problems and over the Pareto-optimal surface for multi-objective problems. The degree of sharing for one individual is calculated by summing a set of sharing function values that indicate the distance between it and the other individuals in the population having the same rank. Fig. 4.4 illustrates the definition of the sharing function for a problem with one objective. The individuals that are close in objective space will have larger imposed sharing function values and those individuals that are far apart will have smaller sharing function values.

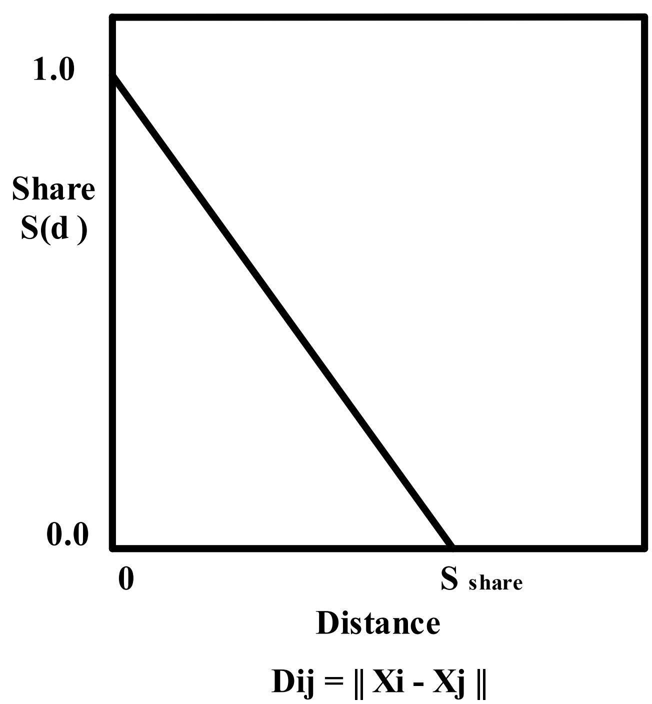


Fig. 4.4. Triangular sharing function

Equation 4.1 presents the equation proposed by Goldberg (1989) to calculate the shared fitness values for each individual in the population.

$$f_s(x_i) = \frac{f(xi)}{\sum_{j=1}^{n} s(d(x_i, x_j))} \tag{4.1}$$

where

n = size of the population

f (x_i) = fitness value of the i^{th} individual

s(d(x_i,x_j)) = sharing value based on distance between i^{th} and j^{th} individual

fs(x_i) = fitness value of i^{th} individual considering the degree of sharing

Equation 4.1 supports the distribution of non-dominated individuals over the Pareto-optimal front and assists in maintain diverse topologies in the population. Fig. 4.5 illustrates the flow of operations performed by the MOGA for Pareto-ranking that implements a sharing function.

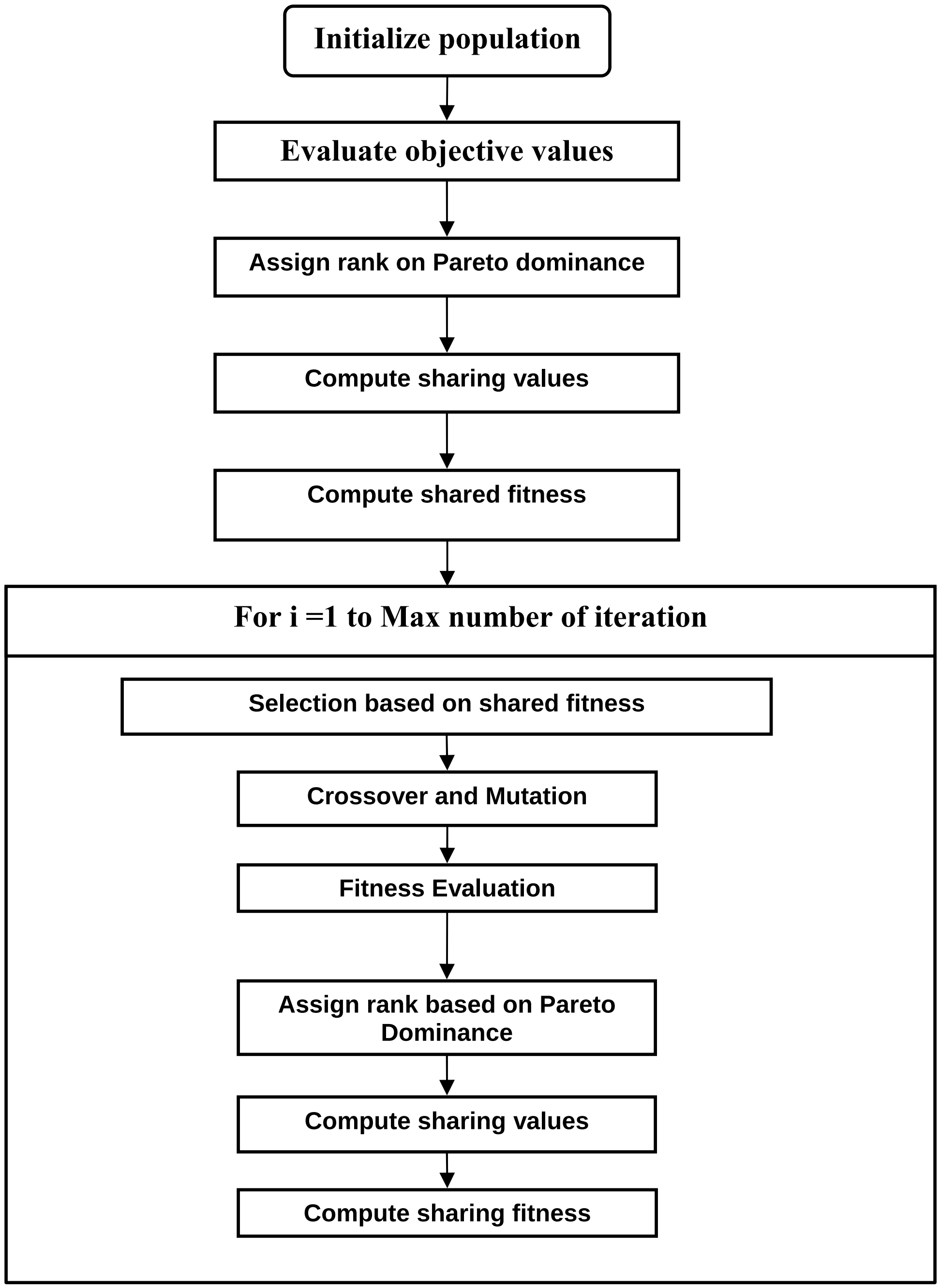


Fig. 4.5. Flow of MOGA based on Pareto-ranking that implements sharing

Optimization based on ranking and sharing function

Multi-objective optimization was performed using ranking and a sharing function to assign fitness values to each individual. In all trials performed, the individuals were selected using tournament selection. The individual in the tournament pool with the lowest rank is always selected. When two individuals are selected having the same rank, then selection is performed based on information provided by the sharing function defined in Equation 4.1. In case the individuals have the same shared fitness value then one individual is randomly selected. The total fitness of each individual is computed using the composite fitness function defined by equation 4.2.

$$F_{TOT} = C - \frac{F_w + F_d}{1 + P_{st}} \tag{4.2}$$

Where

F_{TOT} = Total fitness of each individual evaluated based on the composite fitness function

C = 1000 (Constant used to ensure the generation of only positive fitness)

F_w = Weight fitness

F_d = Displacement fitness

P_{st} = Stress fitness

Equation 4.2 gives the total fitness. In this research maximizing fitness value is sort for. P_{st} is chosen such that it gives the total stress violation. When the stress violation occurs in a member then the stress fitness for that member equals zero. So the overall value of P_{st} becomes less. So, in order to maximize the value of P_{st} all the members in the truss should be within the maximum stress limit. Thus, maximizing stress fitness results in higher value of the total fitness.

In this research the truss weight and deflection were considered as separate objectives. Equation 4.3 state the computation of sharing function values using two objectives that are to be minimized, which are the truss weight and deflection.

$$f_{shared,i} = \frac{F_{TOT,i}}{S_i} \tag{4.3}$$

$$S_i = \sum_{i=1}^{n} C_s - (S_{wt} + S_{dis}) \tag{4.4}$$

$$S_{wt,i} = \frac{F_{wt,i} - F_{wt,j}}{D_{w,MAX}} \tag{4.5}$$

$$S_{dis,i} = \frac{F_{d,i} - F_{d,j}}{D_{d,MAX}} \tag{4.6}$$

Where

$F_{TOT,I}$ = Total fitness values of ith individual

$F_{w,j}$ = Weight of the jth individual

$F_{d,j}$ = Deflection of the jth individual

$D_{w,MAX}$ = Maximum difference in weight in the population

$D_{d,MAX}$ = Maximum difference in deflection in the population

C_s = 1000 (Constant to ensure positive values)

The two objective values for each individual were compared with those of all the other individuals to compute the distances. The summation of the distances for each individual was normalized by the maximum distance, which was determined as the distance between the two individuals that were the farthest away from each other in the population.

The problem search domain for any combinatorial optimization problem, including truss topology and geometry optimization, is extremely large. In addition, a significant portion of the search space is infeasible area that includes unstable or non-optimal trusses that have large deflections or weights. Thus the maximum distance used in the sharing function can cause individuals to continue to explore large infeasible areas. Therefore, the maximum distance calculated in this research in terms of weight and deflection is reduced. The reduction helps in providing a more efficient exploration for design solutions. The maximum distances are reduced by multiplying the maximum distance in terms of weight and deflection with an empirically found constant value.

Equation 4.7 and Equation 4.8 provide the computation of reduced distances in terms of weight and deflection and the empirically determined constant value.

$$D_{w,\mathrm{mod}ified\_MAX} = C_{DW} D_{w,MAX} \tag{4.7}$$

$$D_{d,\mathrm{mod}ified\_MAX} = C_{DD} D_{d,MAX} \tag{4.8}$$

Where

$0 < C_{DW}, C_{DD} \leq 1$,

$C_{DW} = 0.6$,

$C_{DD} = 0.9$.

Results of implementing MOGA for truss optimization

Table 4.1 presents the GA parameters used in this research to find near-optimal truss designs. A number of trials were performed to determine the most effective string length and GA parameters in obtaining consistent performance and quality of designs obtained. Two span lengths (40ft and 60ft) were investigated in the trials presented in this chapter. Fig. 4.6 through Fig. 4.9 present the evolution of trusses for one trial conducted during the 40 ft span length case study over 1000 generations. Fig. 4.10 presents the Pareto-optimal curve (all Rank 1 individuals in the population) obtained considering the tradeoff between satisfying the weight and deflection objectives after 1000 generations were performed. Fig. 4.11 presents the Pareto-optimal curve for weight versus deflection for the 40 ft truss showing all rank 1, rank 2 and rank 3 individuals after 1000 generations. Fig. 4.12 through Fig. 4.15 present a view of how the evolution of trusses was performed for the 60 ft span length case study. Fig. 4.16 presents the Pareto-optimal curve (all Rank 1 individuals in the population) obtained considering the tradeoff between satisfying the weight and deflection objectives for the 60 ft span after 1000 generations. Fig. 4.17 presents a Pareto curve for weight against deflection for a 60ft truss with rank 1 and rank 2 individuals (trusses obtained after 1000th generation).

Table 4.1. Set of GA parameters used in the trials performed for the 40 and 60 ft span problem domain

	40ft	60ft
Generations	1000	1000
Population Size	1000	1000
Crossover Rate	0.8	0.95
Mutation Rate	0.006	0.005
String Length	400	400

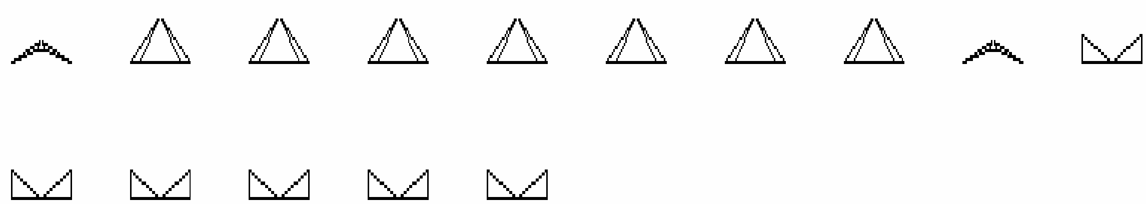


Fig. 4.6. Trusses obtained after the 100th generation for 40-ft span

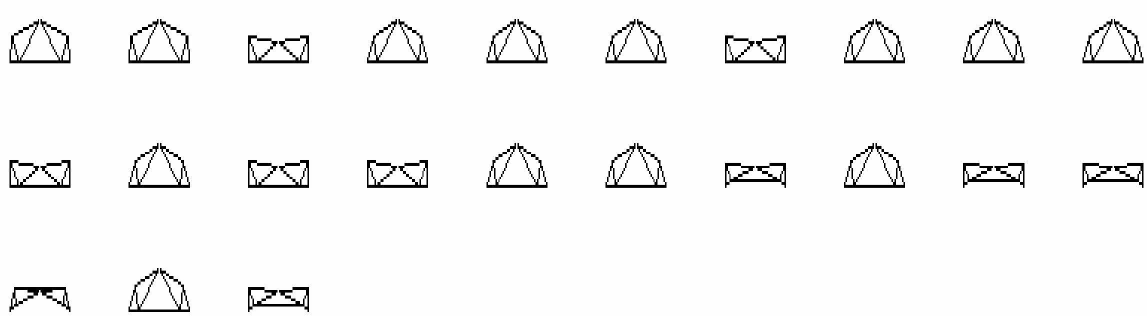


Fig. 4.7. Trusses obtained after the 400th generation for 40-ft span

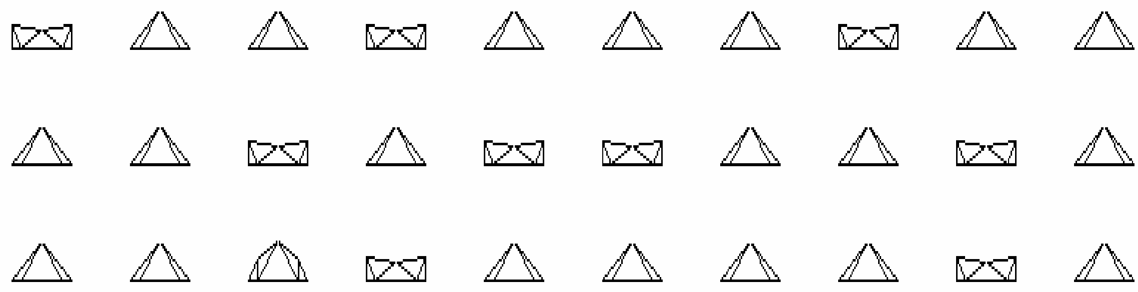


Fig. 4.8. Trusses obtained after the 700th generation for 40-ft span

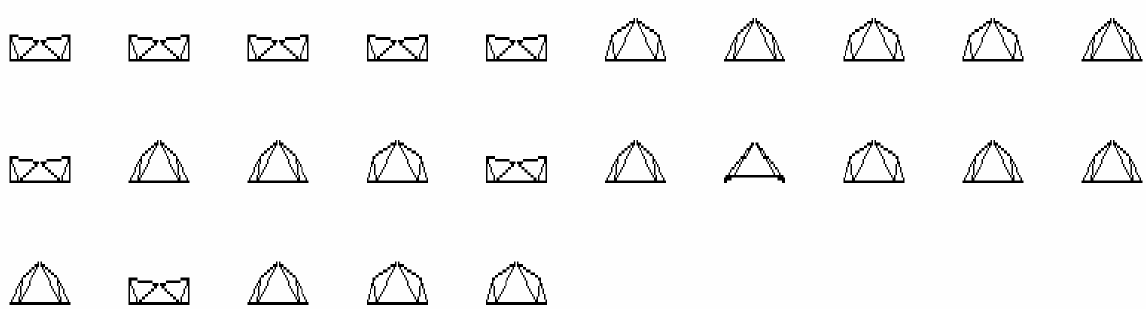


Fig. 4.9. Trusses obtained after the 1000th generation for 40-ft span

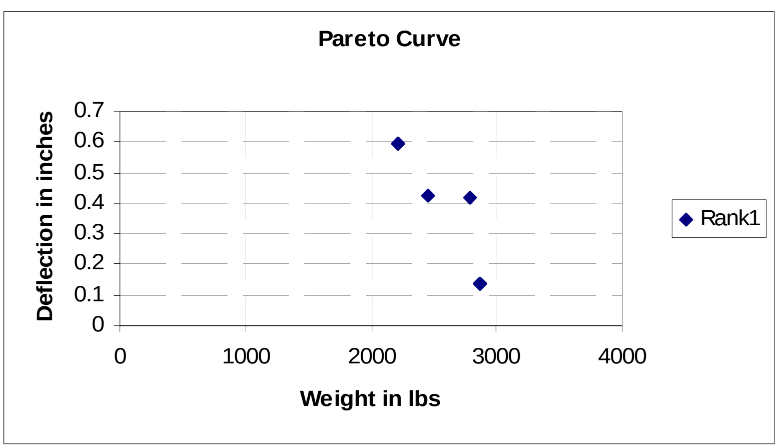


Fig. 4.10. Pareto-optimal front for MOGA trial for 40ft. span (Rank 1 individuals)

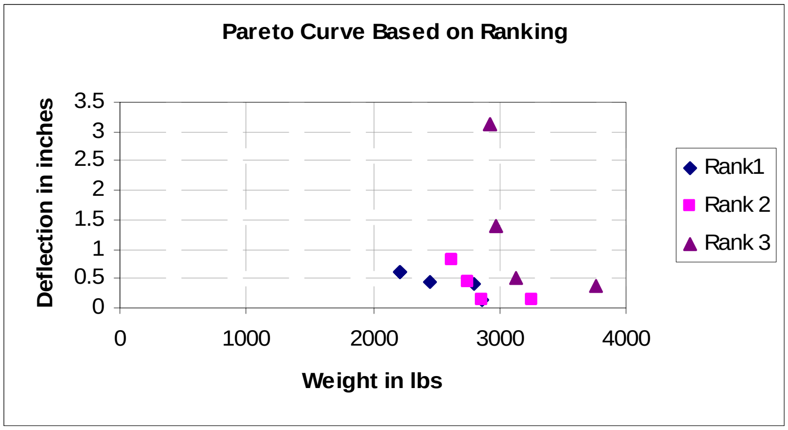


Fig. 4.11. Pareto-optimal front for MOGA trial for 40ft. span (Rank 1, Rank 2, and Rank 3 individuals)

Fig. 4.12. Trusses after the 100th generation for 60-ft span

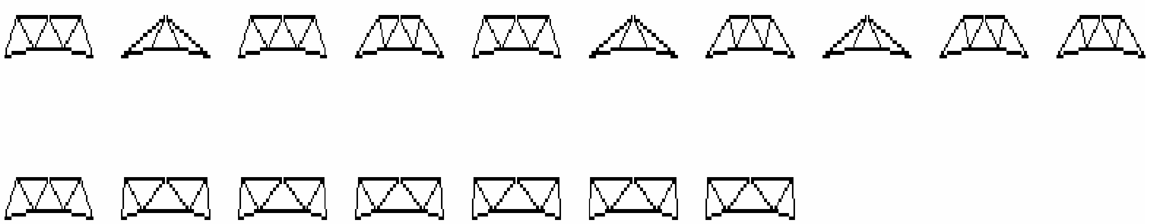


Fig. 4.13. Trusses after the 400th generation for 60-ft span

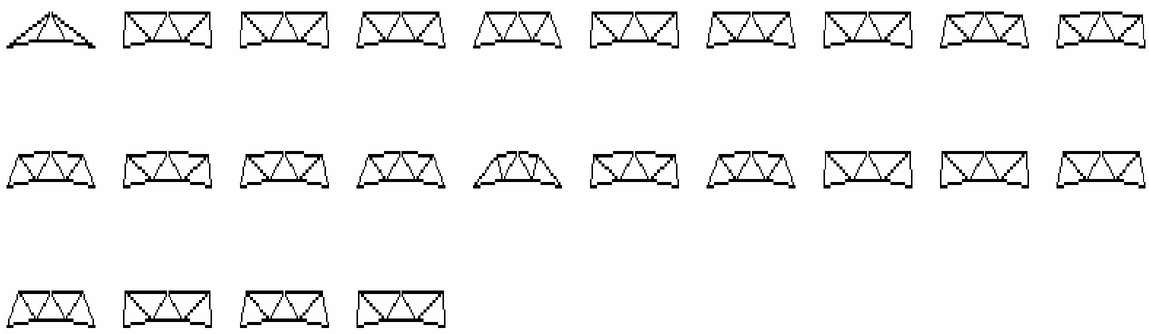


Fig. 4.14. Trusses after the 700th generation for 60-ft span

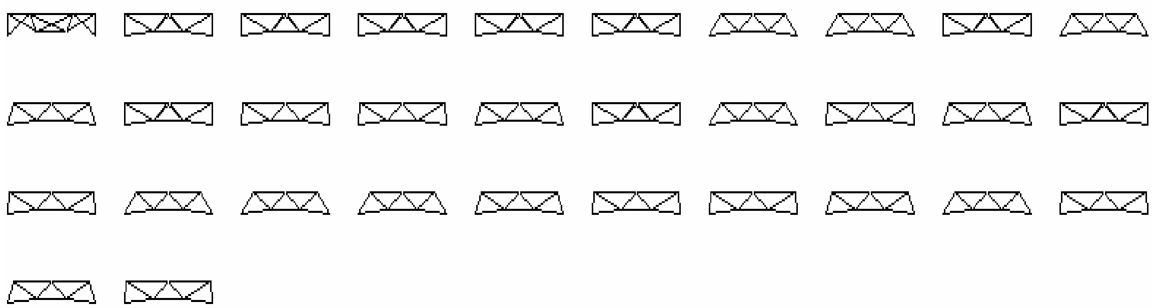


Fig. 4.15. Trusses after the 1000th generation for 60-ft span

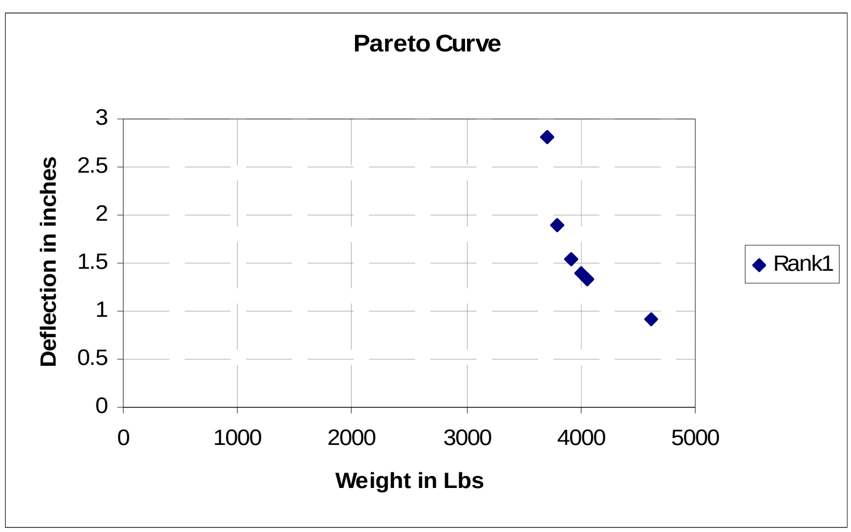


Fig. 4.16. Pareto-optimal front for MOGA trial for 60ft. span (Rank 1 individuals)

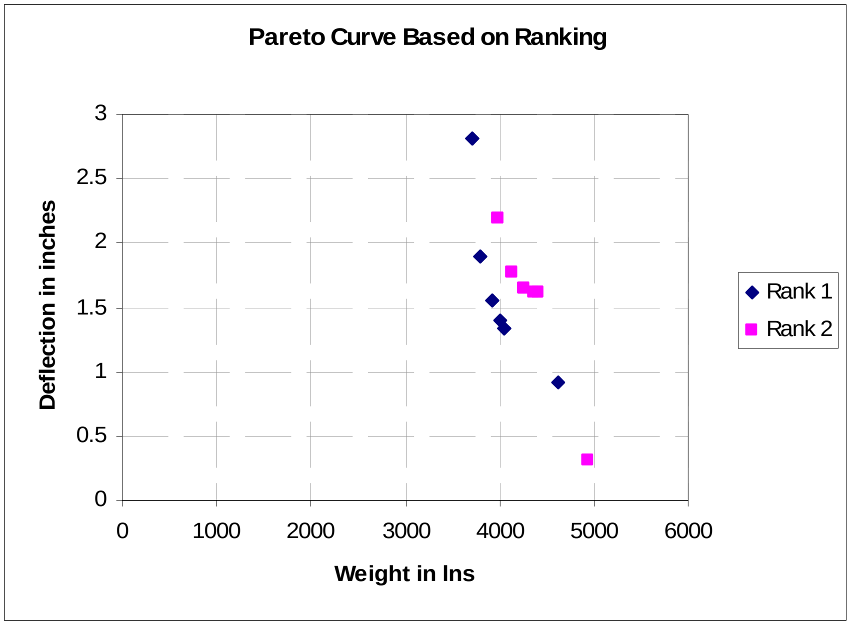


Fig. 4.17. Pareto-optimal front for MOGA trial for 60ft. span (Rank 1 and Rank 2 individuals)

Fig. 4.6 through Fig. 4.9 show how the truss designs evolve by viewing the Rank 1 individuals in the population during different generations for 40ft span. The results indicate that as the number of generations increase there is improvement in topologies and better design alternatives are developed. The same results are shown by the generational views of the 60ft span trial shown in Fig. 4.12 through Fig. 4.15 where there is increase in the number of members through generations, improved topology is found, and the number of feasible design alternatives in the population also increases.

In Fig. 4.10, Fig. 4.11, Fig. 4.16 and Fig. 4.17 the trusses defined on the Pareto-optimal front have very similar topologies, but they differ in the nodal locations and in member section sizes used.

The Pareto-optimal front results in Fig. 4.10 for the 40ft. span showed similarities in truss topologies and were composed of 25 Pareto-optimal trusses even though the population size was 1000. In Fig. 4.10. only shows four dots since there are only four rank one members and the rest twenty one are just a copy.

This means majority of the trusses in the population did not converge to the Pareto-optimal front because of the limited topologies that were optimal. The number of structurally-efficient topologies for the 40-ft span problem domain is expected to be limited due to the short span involved.

In Fig. 4.16, the Pareto-optimal front result for a 60 ft. span trial is presented. The results indicate that the number of Pareto-optimal trusses increases over the number found in the 40ft span trials. In this case, a total number of 41 trusses are obtained. In addition, the number of distinct truss topologies found is greater in the 60ft span case. The results obtained in both the cases (i.e. 40ft and 60ft span) were sensitive to the constants used in the composite fitness functions and the sharing functions and the results were also sensitive to the GA parameters selected.

From the trials performed using the two different span lengths it was difficult to maintain the distributions of individuals along the Pareto-front using the sharing function. This conclusion is supported by the fact that the Pareto-optimal front results reflect that most of the individuals were concentrated in local optimum or most of the individuals were in an infeasible area, which was indicated by trusses that did not meet all of the stress constraints. Most feasible truss designs obtained for both cases showed a lack of diversity in the truss topology. This means that individuals converged to a locally optimum topology. Thus, from the above results it can be interpreted that the MOGA design grammar and representation used with sharing function for this research was not able to effectively search for diverse topologies and reach near-optimal design alternatives within a single trial.

Fig. 4.18 through Fig 4.23 present additional trial results obtained for the 40ft and 60ft span length trials performed.

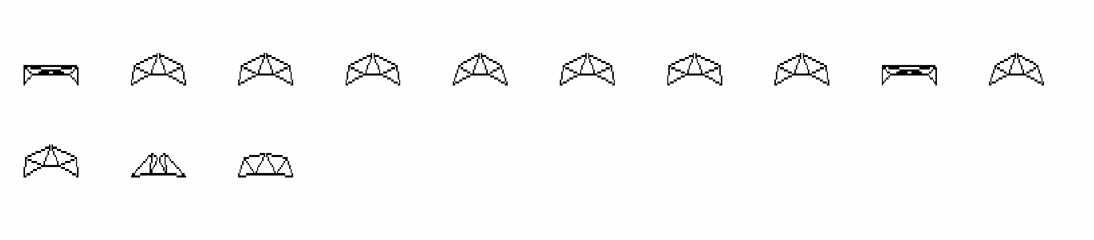


Fig. 4.18. Most optimal individuals after 1000 generations for 40 ft span (Trial 2)

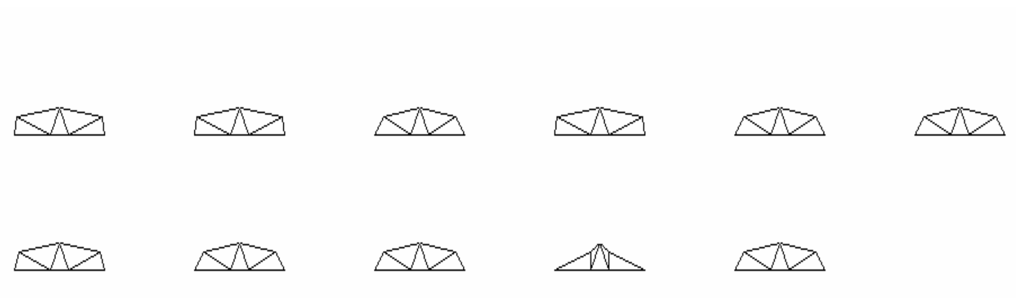


Fig. 4.19. Most optimal individuals after 1000 generations for 40 ft span (Trial 3)

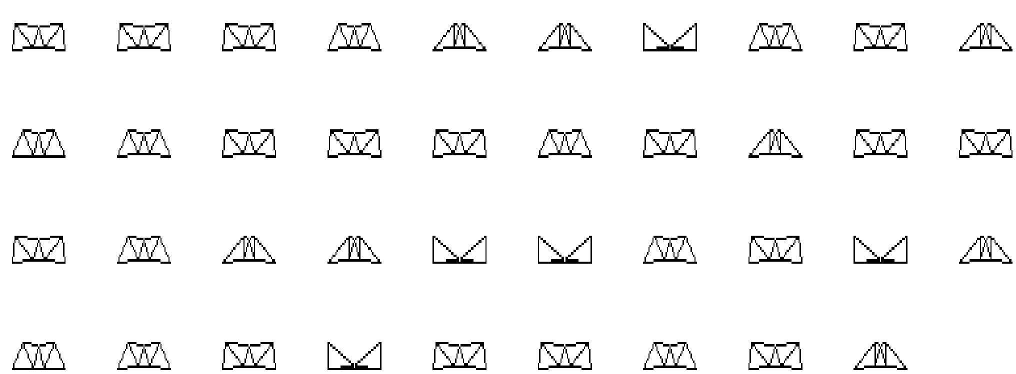


Fig. 4.20. Most optimal individuals after 1000 generations for 40 ft span (Trial 4)

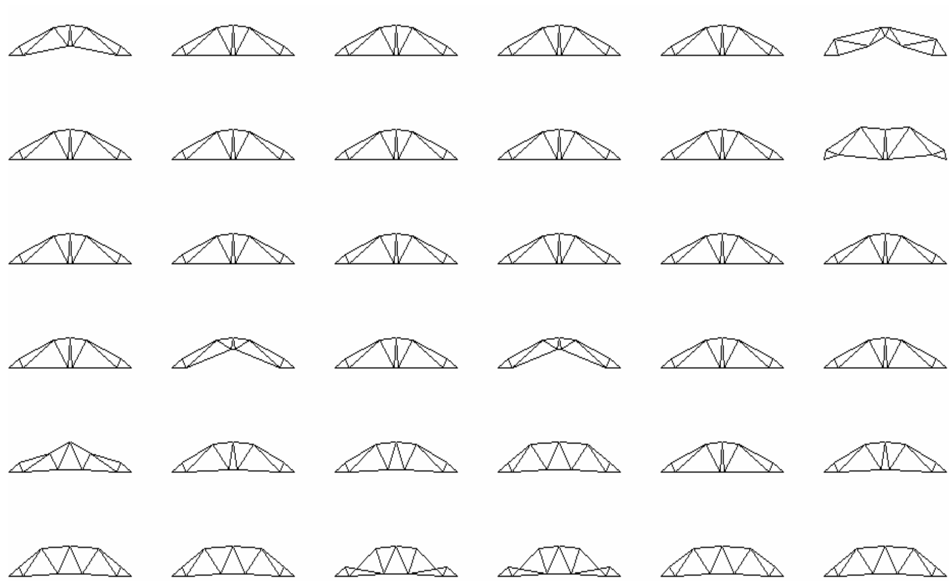


Fig. 4.21. Most optimal individuals after 1000 generations for 60 ft span (Trial 2)

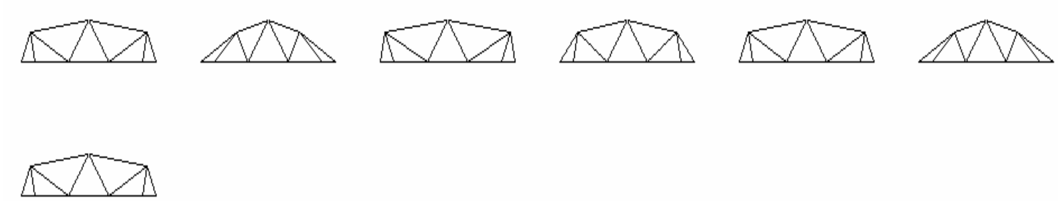


Fig. 4.22. Most optimal individuals after 1000 generations for 60 ft span (Trial 3)

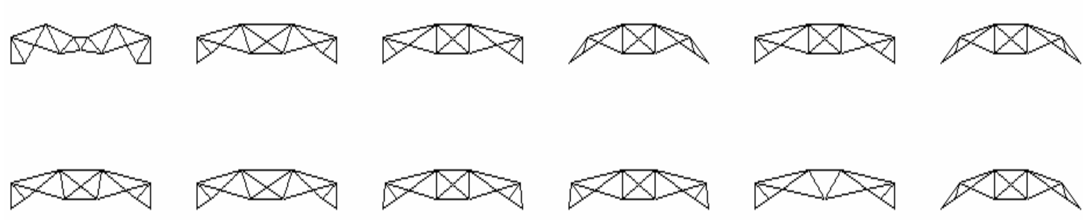


Fig. 4.23. Most optimal individuals after 1000 generations for 60 ft span (Trial 4)

From the results shown in the previous figures, the MOGA method is able to explore different topologies over individual trials. Through performing different trials (in an optimization sense, different starting points), the MOGA is able to propose diverse topologies to the user. Information about trusses in Fig. 4.20 is presented in APPENDIX I. And the information about trusses in Fig. 4.23 is presented in APPENDIX II.

CHAPTER V
EVALUATION OF THE PROPOSED MOGA METHOD AND REPRESENTATION

This section of the thesis discusses the definition of the benchmark problem and results obtained from the proposed MOGA method and representation on the benchmark problem. The results obtained in this research are compared with results obtained previously by other researchers on the same benchmark problem. To validate the proposed method and representation, the benchmark problem defined in the paper "Evolution of optimum structural shapes using genetic algorithm" by Shrestha and Ghaboussi (1998) was studied then the algorithm was modified to meet the predefined conditions in the benchmark problem and the results obtained were compared to the benchmark problem.

Benchmark problem definition

Problem domain description

The methodology proposed by Shrestha and Ghaboussi (1998) introduces the concept of physical design space, which is the specified physical space within which the generated structure is fully enclosed. This space can have any arbitrary contiguous shape and may contain internal "hole" through which no part of the generated structure must pass. The physical design space allows limits to be imposed on the shape of the generated structure, which is important to meet functionality considerations.

The evolved structure can acquire any shape within the physical design space. Some important features of the evolved structures are listed below:

- The structure can have any number of free nodes. It can also have specified partially fixed or fixed nodes, some of which may be loaded or support nodes. The free nodes can occupy any position within the physical design space, whereas the partially fixed nodes will have some of their nodal co-ordinates specified. Similarly, the structure can have any number of members, with any pattern of nodal connectivity. It can also have members whose end nodes, which must be either fixed or partially fixed, are specified. The partially fixed and fixed nodes and the fixed members, if specified must be present in all the generated structures.

- The members are chosen from a set of discrete member sizes, assumed to be a subset of commercially available member sizes from standard design manuals.
- The structure may be subjected to either static or moving, single or multiple loadings. They may include self weight of the structure, various live loads, or any other loadings required by the design codes.

In addition to the above, the generated structure can contain truss or beam elements or both. The supports can either be fixed, roller, or pinned.

In the previous research study on the benchmark problem, the method applied addressed various design objectives and constraints. The design constraints were classified into member and nodal constraints. In the results presented in Shresta and Ghaboussi, weight minimization was considered as the design objective; the member constraints considered are stress, slenderness ratio, minimum member length, maximum member length, member symmetry; and the nodal constraints considered are the nodal displacements and nodal symmetry. Table 5.1 provides details of member and nodal constraints used by Shresta and Ghaboussi.

Problem description

The methodology proposed by Shresta and Ghaboussi aimed at attaining a minimum weight optimal truss shape with simple supports. The span lengths are set at 70 meter (about 230 ft). The maximum height of the structure was 10 meter (about 33 ft). The generated structures could contain any number of free nodes and any number of members. The structural elements are specified to be truss elements. The members are selected from a set consisting of 27 standard AISC sections (1998), ranging from W 14 * 22 through W 14 * 426. The material properties are those of steel (E = 2.01*10^5 MPa, f_y = 248.8 MPa, ρ = 7.85103 Kg/ m^3). The relevant AISC ASD design specifications are followed.

The specified allowable values are: $\sigma_a = 0.6^* f_y$; $s_a^T = 300$; $s_a^C = 200$. The allowable compression stress σ_{aj}^C, is determined from buckling considerations, based on code requirements. The displacements are limited to $(1/1000)^{th}$ of the span length, in accordance with relevant AASHTO specifications (1989). Fig 5.1 provides a figure representing the design space, the boundary conditions, and the loading used in the example problem.

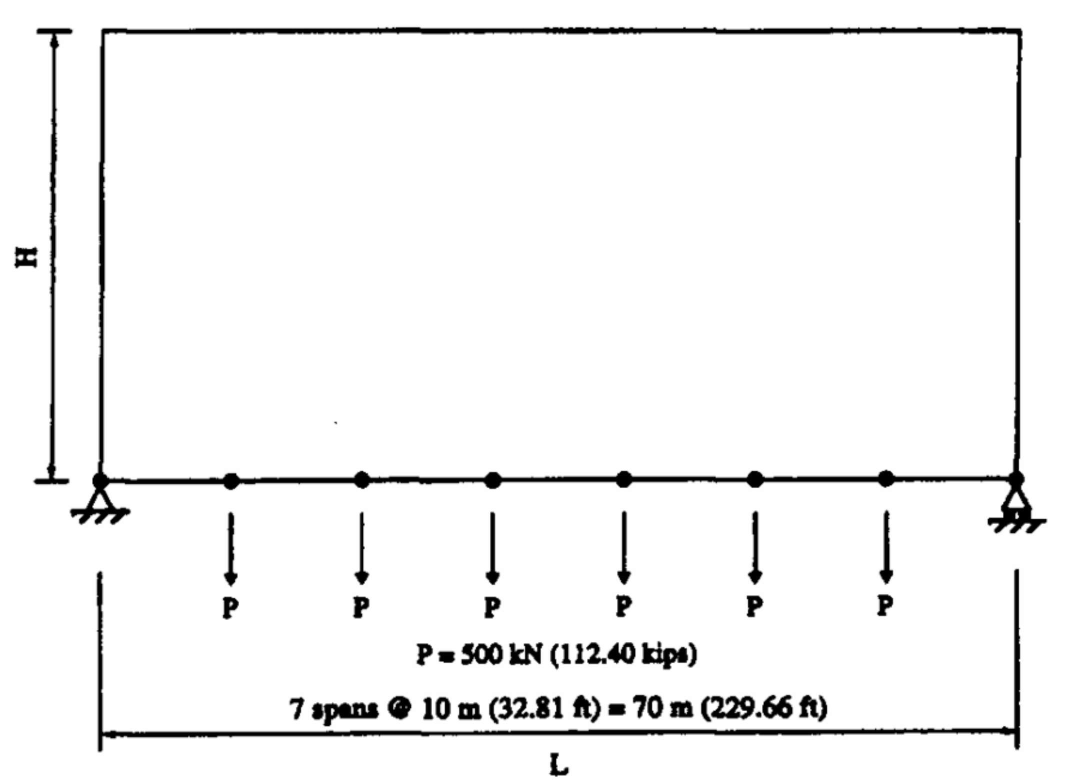


Fig. 5.1. Physical design space, boundary condition and loading for the benchmark problem (Shresta and Ghaboussi, 1998)

The total load applied to the trusses in the benchmark problem is 674.4 Kips and the span 'L' used in the problem is 70m (230ft). The height of the structure 'H' is 10m (33ft).

In the previous study, the population size was limited to 100. The mutation rate is 0.002 and the crossover rate is 1.0. Multipoint crossover is used with the number of crossover sites ranging from 2 to 10 chosen randomly. The number of generations performed was 10,000.

Table 5.1. Benchmark problem results (Shreseta and Ghaboussi, 1998)

Truss Weight (kgs)	Generation attained	Truss Deflection (inches)
60,329	9754	2.753

Fig. 5.2 represents the development of the optimal truss topology and geometry for the benchmark problem through different generation and the final attainment of the best optimal truss (Shresta and Ghaboussi, 1998). Table 5.1 presents the optimal truss weight obtained in the benchmark problem and the maximum deflection.

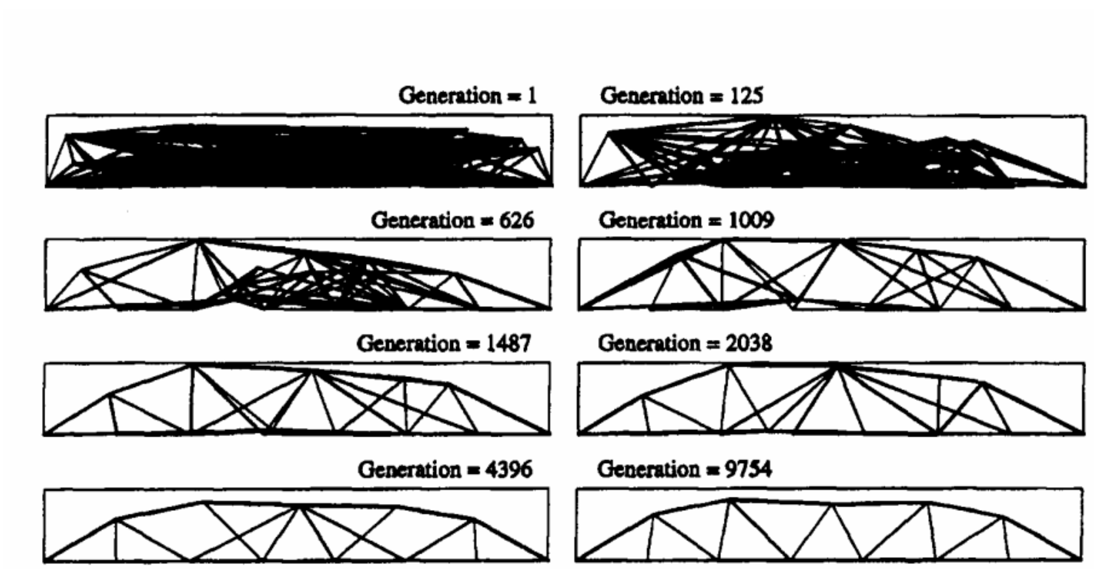


Fig. 5.2. Development of the truss structure through different generation and the final attainment of the most optimal truss (Shresta and Ghaboussi, 1998)

Evaluation of the proposed MOGA method and representation by the benchmark problem

To evaluate the developed MOGA method, the same problem domain as used in the benchmark problem was used with the same loading conditions and set of member sizes. Table 5.2 through Table 5.5 present the optimal truss weight and deflection obtained from different trials using the proposed MOGA methodology along with the GA parameters used in the problem.

Fig. 5.3 through Fig. 5.10 present the final population of near-optimal truss designs obtained after 1000 generations from different trials. The majority of the truss designs obtained in the final generation are similar in topology in the trials performed. These trusses do show more variation in their geometry (nodal locations) and in the member sections sizes assigned.

Table 5.2. Parameters used in trial 1 and the most optimal results obtained

Number of Generations	Crossover Rate	Mutation Rate	Truss Weight (kgs)	Generation attained	Truss Deflection (inches)
1000	0.8	0.005	95417.69	1000	3.14

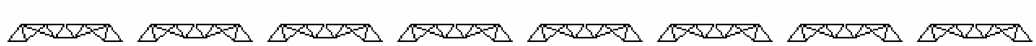


Fig. 5.3. Truss topologies obtained after the 1000^{th} generation in trial 1

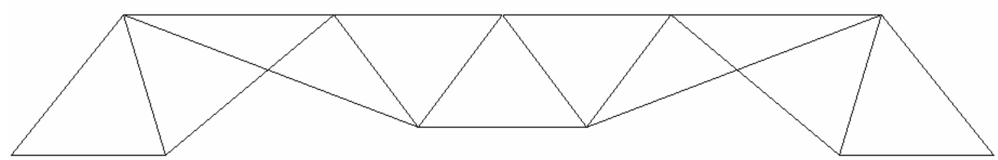


Fig. 5.4. Fittest truss design obtained after 1000th generation in trial 1

Table 5.3. Parameters used in trial 2 and the most optimal results obtained from the developed method

Number of Generations	Crossover rate	Mutation Rate	Truss Weight (kgs)	Generation attained	Truss Deflection (inches)
1000	0.8	0.0045	71985.43	1000	3.74

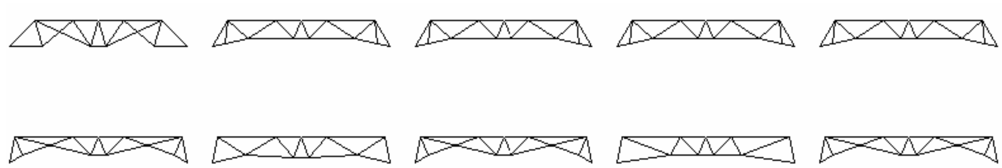


Fig. 5.5. Truss topologies obtained after the 1000th generation in trial 2

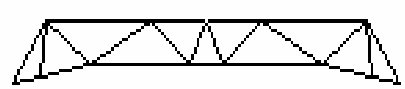


Fig. 5.6. Fittest truss design obtained after 1000th generation in trial 2

Table 5.4. Parameters used in trial 3 and the most optimal results obtained from the developed method

Number of Generation	Crossover Rate	Mutation Rate	Truss Weight (kgs)	Generation attained	Truss Deflection (inches)
1000	0.8	0.004	71534.3	1000	2.598310

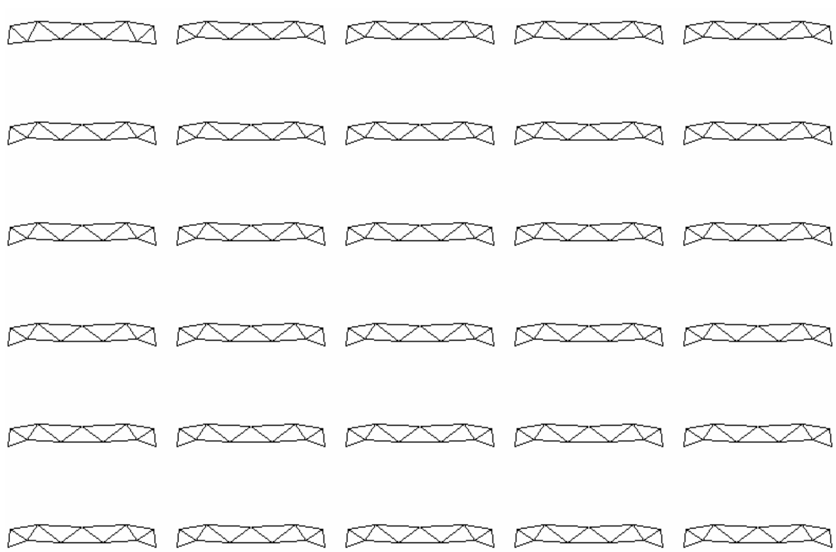


Fig. 5.7. Truss topologies obtained after the 1000th generation in trial 3

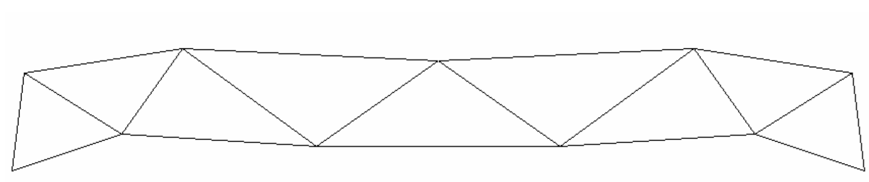


Fig. 5.8. Fittest truss design obtained after 1000th generation in trial 3

Table 5.5. Parameters used in trial 4 and the most optimal results obtained from the developed method

Number of Generation	Crossover Rate	Mutation Rate	Truss Weight (kgs)	Generation attained	Truss Deflection (inches)
1000	0.8	0.003	68079.13	1000	3.15

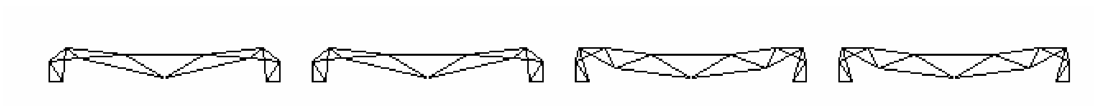


Fig. 5.9. Truss topologies obtained after the 1000^{th} generation in trial 4

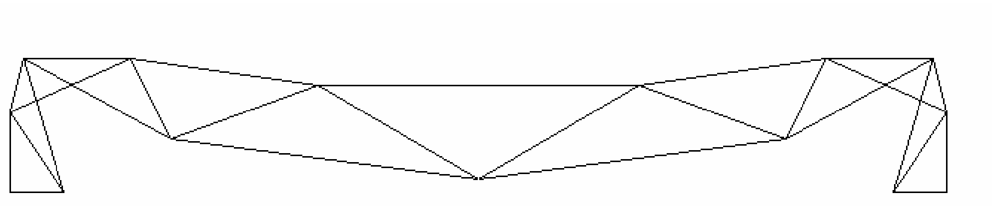


Fig. 5.10. Fittest truss design obtained after 1000^{th} generation in trial 4

Based on the results obtained from the four trials, the best truss design obtained with respect to weight displacement and stress in the members is obtained in trial 3.

From the results presented in Table 5.1 through Table 5.4, it can be concluded that the truss designs obtained using the proposed MOGA method have larger weight, while the deflection is under the permissible limits used in the example problem (1/1000 of the span). Fig. 5.8 presents the most optimal truss obtained by MOGA method in four trials. It should be noted that the computational time taken to obtain the truss designs using the proposed MOGA method is less compared to the result obtained by Shresta and Ghaboussi (1998), since the optimal solution is attained within fewer generations.

In order to validate the results obtained from the proposed MOGA method, a separate structural analysis was performed. The same stress and deflections were obtained in both programs. Table 5.6 presents the results obtained from structural analysis package compared to those obtained from the proposed MOGA method. There is a slight variation of the results shown in Table 5.6 due to the conversion of units from meters to feet. Fig. 5.11 presents a view of the truss examined by the structural analysis package.

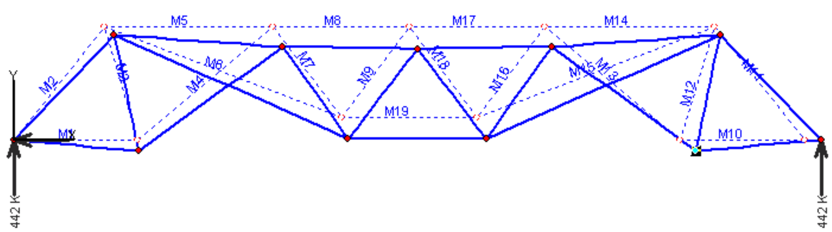


Fig. 5.11. View of the truss design provided from the structural analysis package

Table 5.6. Comparison of the member stresses between the analysis package and the MOGA method

Member	Structural Analysis Package	Proposed MOGA method
1	11.9269	11.2701
2	6.0786	6.724581
3	15.548	15.03081
4	4.3067	4.390468
5	23.223	22.52593
6	12.4418	12.95195

Table 5.6. Continued

Member	Structural Analysis Package	Proposed MOGA method
7	3.1937	3.275068
8	13.615	13.09089
9	1.0647	1.000003
10	11.9269	11.2701
11	6.0786	6.724581
12	15.548	15.03081
13	4.3067	4.390469
14	23.223	22.52593
15	12.4418	12.95195
16	3.1937	3.275069
17	13.615	13.09089
18	1.0647	1.000003
19	42.9339	41.8353

Due to the large search space of potential design alternatives defined by the unstructured problem domain, the optimization of the member section sizes is not performed to its fullest extent. Instead, the optimization of truss topology and geometry is performed to a greater extent. In addition, once the population begins to converge it is difficult to continue member size optimization in any MOGA. In order to determine the state of member size optimization performed by the MOGA method, local optimization was performed on the truss design produced by the MOGA method in order to determine how much weight savings could be obtained. This investigation was carried out with the trusses obtained in trial 1 of the benchmark problem.

Trial 1 is selected for optimization, to show to what extent the trusses can be optimized using fine tuning since it has the highest weight. And from the results it can be seen that it can be optimized to like twenty thousand Kgs.

Table 5.7 presents the changes made to the truss obtained by the MOGA method to locally optimize the member section sizes and Table 5.8 presents the final structural analysis results obtained after local optimization was performed. Fig 5.12 shows the result view of the locally optimized truss. The results should in Table 5.8 confirm that there is a drastic reduction in weight of the truss, although this comes with a slight increase in truss deflection. Also, it can be concluded from the results shown in Table 5.7 that only four different member section sizes result from the local optimization of member size, which makes the structure more practical from a constructability viewpoint also.

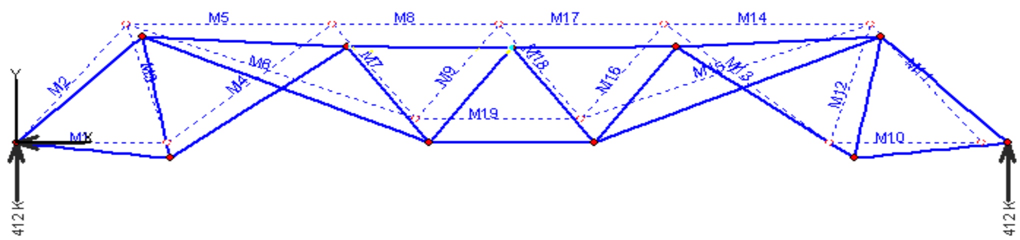


Fig. 5.12. View of the truss design undergoing local member section size optimization

Table 5.7. Results obtained through local member size optimization for fixed topology

Member	Original Section Size	Optimized Section Size	Original Truss Weight	Optimized Truss Weight
M1	W14x99	W14x34	3.5727	1.2277
M2	W14x311	W14x99	13.064	4.1593
M3	W14x34	W14x26	1.1653	0.8961
M4	W14x311	W14x34	15.9349	1.7434
M5	W14x257	W14x257	12.6567	12.6567
M6	W14x426	W14x426	31.3519	31.3519

Table 5.7. Continued

Member	Original Section Size	Optimized Section Size	Original Truss Weight	Optimized Truss Weight
M7	W14x370	W14x99	12.1656	3.2479
M8	W14x311	W14x257	12.2415	10.1254
M9	W14x26	W14x26	0.8583	0.8583
M10	W14x99	W14x34	3.5727	1.2277
M11	W14x311	W14x99	13.064	4.1593
M12	W14x34	W14x26	1.1653	0.8961
M13	W14x311	W14x34	15.9349	1.7434
M14	W14x257	W14x257	12.6567	12.6567
M15	W14x426	W14x426	31.3519	31.3519
M16	W14x370	W14x99	12.1656	3.2479
M17	W14x311	W14x257	12.2415	10.1254
M18	W14x26	W14x26	0.8583	0.8583
M19	W14x99	W14x426	3.8975	16.7417

Table 5.8. Comparison of results obtained through local member size optimization for fixed topology with the results from the developed methodology

Original Truss Weight (kgs)	Optimized Truss Weight (kgs)	Original Truss Deflection (inch)	Optimized Truss Deflection(inch
95417.69	67781.82	3.14	3.96

The individuals in each of the regions circled below in Fig. 5.13. have mostly the same topology and geometry. They vary in their member sizes to some extent. Thus, each of the sub curves is the Pareto-optimal curve that would be obtained when looked at a restricted design space – the topology and geometry fixed or very similar and only changes in member sizes perhaps.

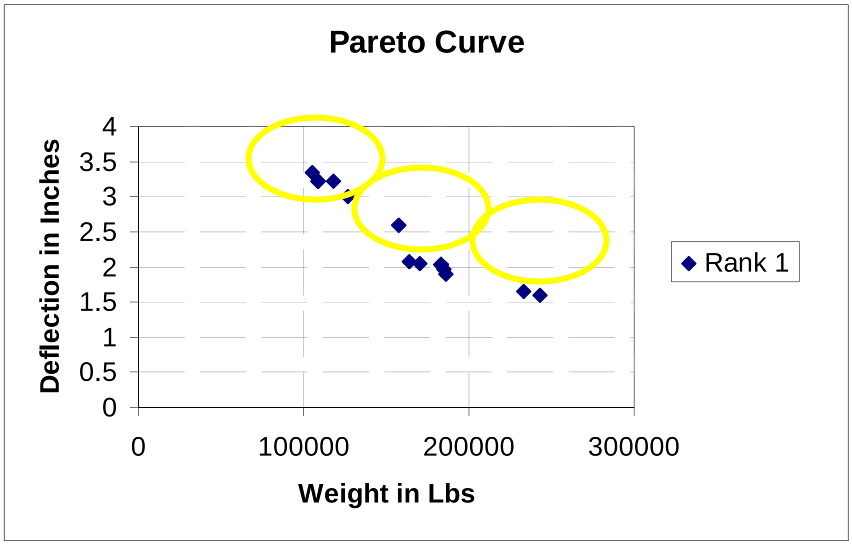


Fig. 5.13. Pareto-optimal curve representing rank 1 individuals obtained in trial 3

Table 5.9. Comparison table showing the most optimal weights attained

Generation Attained in example problem	Generation attained by the developed methodology	Most optimal weight attained by the example problem (Kgs)	Most optimal weight attained by the developed methodology (Kgs)
10000	1000	60,329	71,534.3

Table 5.9 presents a comparison table between the most optimal weight attained and the generations taken to attain the most optimal solution by Shresta and Ghaboussi (1998) and the proposed MOGA method. Although the truss weight attained by the MOGA method is higher, as shown previously, local optimization of member sizes can be applied to reduce it substantially. The major advantage of the proposed MOGA method is that it generates trusses randomly and does not start from any ground structure, so a wide variety of trusses can be explored and the final results can be locally optimized to obtain feasible truss designs while also maintaining deflections within the limits. Moreover the generation of attainment of the most optimal individual in the proposed methodology is pretty less which implies that the computational time taken would be less compared to the example problem.

Fig. 5.14 through Fig. 5.17 present the Pareto-optimal trusses found from additional trials. These results shown in these figures indicate that the program performs an extensive search of the problem domain and is capable of exploring diverse truss topologies although in different trials, not in a single trial. The MOGA parameter values used in each trial plays an important part along with the initial random seed to determine the type of trusses evolved and the final population.

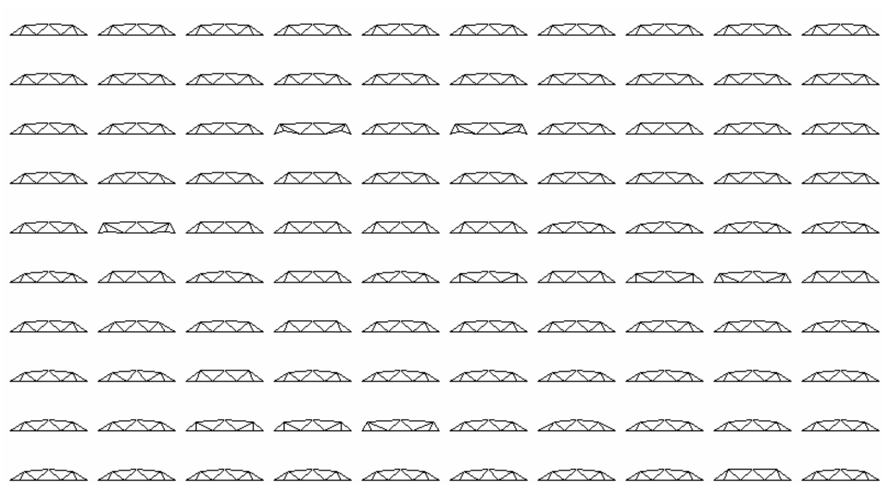


Fig. 5.14. Pareto-optimal truss topologies obtained from trial 5

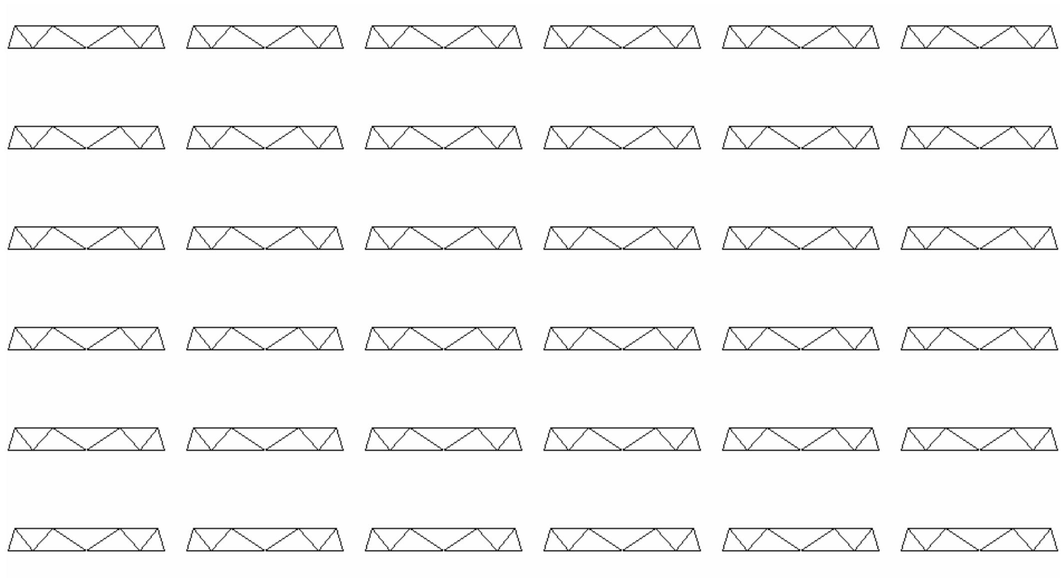


Fig. 5.15. Pareto-optimal truss topologies obtained from trial 6

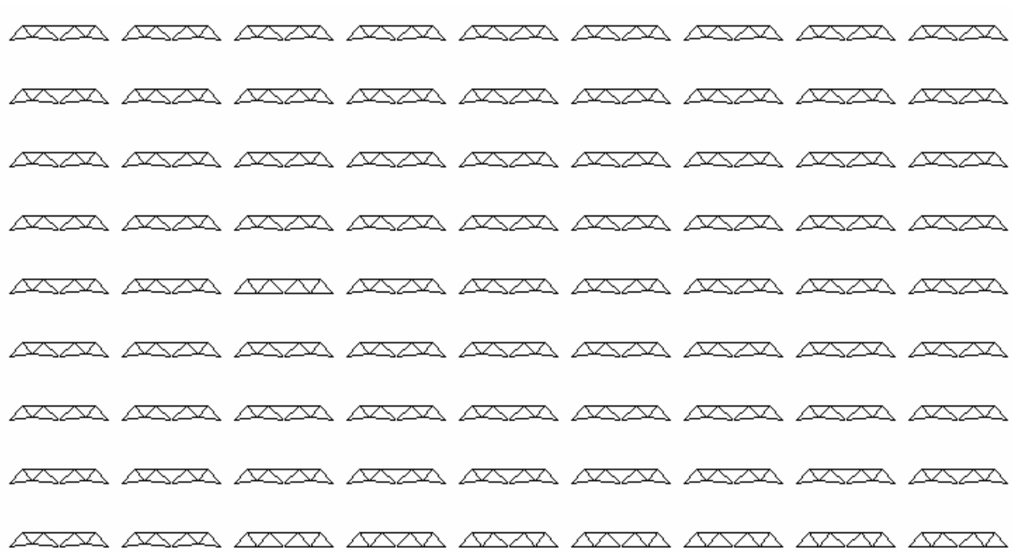


Fig. 5.16. Pareto-optimal truss topologies obtained from trial 7

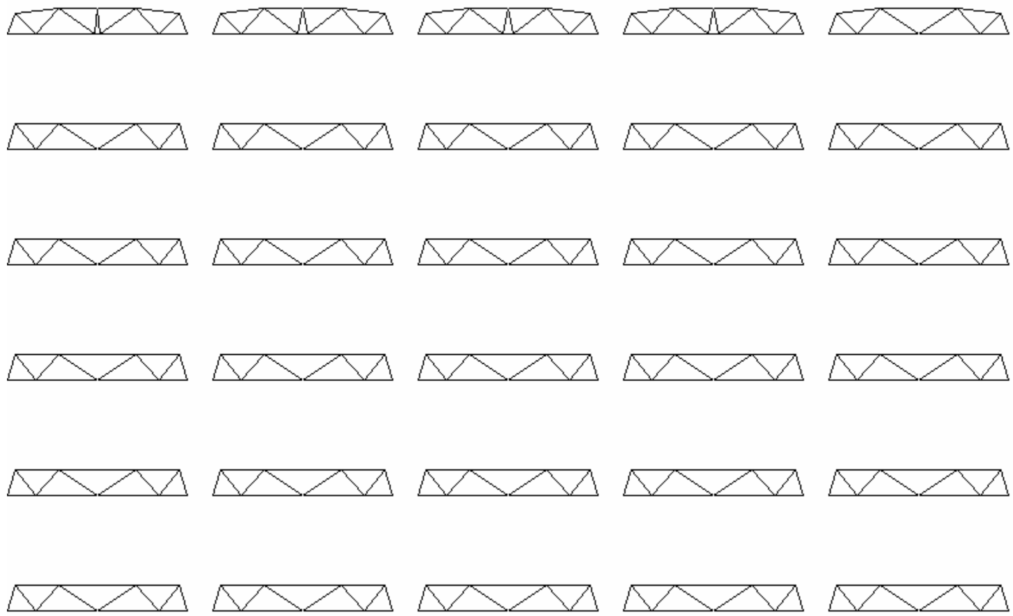


Fig. 5.17. Pareto-optimal truss topologies obtained from trial 8

Although the propose MOGA method does perform a broad search to come up with truss topologies, it is clear from Fig. 5.3 through Fig. 5.10 that in the final population obtained in each trial the topologies were very similar. Different topologies were obtained after running several trials. The Pareto front presented in Fig. 5.13 (trial 3) reflects that most of the individuals were concentrated in local optimum or most of the individuals were in the infeasible area. Since most of the feasible trusses showed lack of diversity in truss topologies, this implies that individuals converged to a locally optimum topology. The selection of the GA parameters also plays a role in the generation of the trusses making up the Pareto-optimal front. The proposed MOGA method is capable of evolving trusses that have lower weight and deflections, but the design representation is not fully effective in providing support for the simultaneous optimization of topology, geometry, and member size in a single trial.

CHAPTER VI
CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

Summary of objectives

The objective of this research was to develop an effective genetic algorithm design representation for multi-objective truss optimization that can support the design a set of near-optimal trusses. To meet this objective, several MOGA implementing various diversity preserving strategies were investigated.

As a first step, different design grammars were investigated based on their ability to produce diverse truss designs and the best one was selected. Then an investigation was performed to determine the optimal MOGA parameter settings through trials. As discussed in Chapter IV, several trials were performed to determine the optimal string length. In addition, several criteria discussed in Chapter III were used to help understand the characteristics of the unstructured problem domain defined in this research.

The definition of a large, complex problem domain made it difficult for the individuals in the GA population to explore the search space and to simultaneously optimize the truss topology, geometry, and member sizes. Several criteria discussed in chapter IV were implemented and the GA parameters were very carefully chosen to preserve the quality of the Pareto-optimal front and to try to maintain diversity in the population. In the proposed algorithm, a niche count was computed using a sharing function (Goldberg 1989). The sharing function helped to maintain some diversity in the population. Sharing is formulated in terms of how close the trusses are in objective space –weight and deflection

To evaluate the performance of the proposed MOGA method, trials were performed on a benchmark truss problem domain and the results obtained were compared with result obtained by other researchers (Shrestha and Ghaboussi 1998) considering all the criteria defined in the benchmark problem. The results of the comparison showed that the performance of the proposed MOGA method, in which the sizing, shape and topology were simultaneously performed, was effective in evolving a variety of truss topologies compared to previous published results, which evolved from a ground structure. However, the diverse topologies were obtained over several trials not within the same population in a single trial. In addition, the proposed MOGA method was not able to locally optimize the member section sizes. This research effort investigated the effect on truss weight and deflection obtained by locally optimizing the member section sizes for a given truss topology and geometry. The results indicate that the significant weight reduction is achieved by additional local optimization.

Future work

The main purpose of the representation and algorithm proposed in this research is to support the development of an efficient computational model that is capable of generating diverse truss topologies and geometries. However, the generation capability is constrained to some extent in the formulation by the specific design grammar used. In addition, the large unstructured problem domain defined increase the size of the search space. Based on results obtained in this research, the size of the feasible area in the search space is very small. In addition, the trusses on the Pareto front obtained in this research are not optimal compared to trusses designed by engineering practice to some extent. Therefore even though the trusses are well optimized based on deflection and weight they can not be put for practical use. The truss structures can be made practical by using fewer member sizes, but the deflection and weight constraints may not be satisfied. Thus, all the problems of making the truss structure practical and keeping the weight and deflection criteria within limit fine tuning (sizing optimization) can be done to the final results obtained from the proposed method.

To address the problems that were raised during this research effort, the following research directions are recommended:

- *Develop an efficient practical design grammar and representation*

  Design grammar plays an important role in the whole process of development of the computational model. Even though in this research the design grammar investigated was able to examine topologies and geometries, it was not capable of effectively optimizing truss designs. Looking at the research literature, many design grammars have been proposed. The design grammars involving some predefined shapes like triangles, rectangles etc. can be used as they would not produce any unstable trusses. Moreover the search space will be considerably reduced. However, the flexibility of the design grammar must be considered in order to obtain near – global optimal set. Having too small of a search space will prevent the MOGA from synthesizing the design alternatives effectively.

- *Use parallel computing on a UNIX based system*

  Parallel computing on a UNIX based system can be used to reduce the computational time.

- *Backup pool*

  To prevent the trusses from getting stuck in local optimum, a back-up pool may be considered that stores the best trusses from a generation and they don't come into play when the selection of the individuals are done for the next generation. The individuals stored in the backup pool may then be allowed to undergo crossover and mutation with the other selected individuals to some extent. This may prevent individuals from getting stuck in some local optimum.

- *Fine tuning*

  The most important criteria to make a feasible truss structure are to have consistency in member areas. A twelve member truss cannot afford to have twelve different member sizes. So, by doing fine tuning (sizing optimization) to the results obtained, some feasible member sizes can be selected and the truss structure can be made practical along with meeting all the weight and deflection criteria.

In this research, investigation of the MOGA design representation was performed. The future improvement of the design grammar and the computing methods will allow the evolution of optimized truss designs for practical use. In addition, other soft computing methods like neural networks can be integrated to help support intelligent designs in the future.

REFERENCES

AISC (2001). *Manual of Steel Construction: Load and Resistance Factor Design*, American Institute of Steel Construction, Chicago, IL.

Berke, L., and Khot, N. S. (1987). "Structural optimization using optimality criteria." *Computer Aided Optimal Design: Structural and Mechanical System,* Berlin, Springer-Verlag, 235-269.

Deb, K. and Gulati, S. (2001). "Design of truss-structures for minimum weight using genetic algorithms." *Finite Elements in Analysis and Design*, 37, 447-465.

Gage, P.J., Kroo, I.M., and Sobieski, I.P. (1995). "Variable-complexity genetic algorithm for topological design." *AIAA J.*, 33(11), 2212-2217.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.

Hajela, P. and Lee, E. (1995). "Genetic algorithm in truss topological optimization." *International Journal of Solids and Structures*, 32(22), 3341-3357.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Univ. of Michigan Press: Ann Arbor. Reprinted in 1992 by MIT Press, Cambridge MA.

Jenkins, W.M. (1997). "On the application of natural algorithms to structural design optimization." *Engineering Structures,* 19(4), 302-308.

Jenkins, W.M. (2001). "A decimal-coded evolutionary algorithm for constrained optimization." *Computers and Structures*, 80, 471-480.

John, K. V., and Ramakrishnan, C. V., P. (1987). “Minimum weight design of trusses using improved move limit method of sequential linear-programming..”*Computers and Structures,* Pergamon-Elsevier Science Ltd., Vol.27 Issue. 5 pp.583-591 1987.

Lin C-Y, and Hajela, P. (1992). “Genetic Algorithm in Optimization Problems with Discrete and Integer Design Variables.”*Eng. Opt.,* Vol. 19,309-327.

Lin C-Y, and Hajela, P. (1993). “Genetic Search Strategies in Large Scale Optimization.”*Proceedings 34th ALAA/ASME/ASCE/ASC SDM Conf.,* La Jolla, Ca, ASCE, New York.NY, 2437-2447.

Lin C-Y, and Hajela, P. (1994). “Design optimization with advanced genetic search strategies.” *J Adv Eng Softw* 21:179–189.

Raich, A.M. (1999). *An evolutionary based methodology for representing and evolving structural design solutions*, Ph.D. thesis, Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign.

Rajan, S.D. (1995). “Sizing, shape, and topology design optimization of trusses using genetic algorithm.” *Journal of Structural Engineering*, 121(10), 1480-1487.

Rajeev, S. and Krishnamoorthy, C.S. (1992) “Discrete optimization of structures using genetic algorithms.” *Journal of Structural Engineering*, 118(5), 1233-1250.

Rajeev, S. and Krishnamoorthy, C.S. (1997). “Genetic algorithms-based methodologies for design optimization of trusses.” *Journal of Structural Engineering*, 123(3), 350-358.

Rechenberg, I. (1965). “Cybernetic solution path of an experimental problem.” *Farnborough*, UK: Royal Aircraft Establishment, 2437-2447, Vol. Library Translation 1122.

Sakamoto, J., and Oda, (1993). “Technique for optimal layout design for truss structures using genetic algorithms.” *Proceedings 34th ALAA/ASME/ASCE/ASC Structural Dynamics and Material Conference ALAA/ASME Adaptive Structures Forum*, New York, NY, 2402-2408.

Schmit, L. A. (1981). “Structural synthesis- its genesis and development.” *AAAI Journal, 19*(10), 1249-1263.

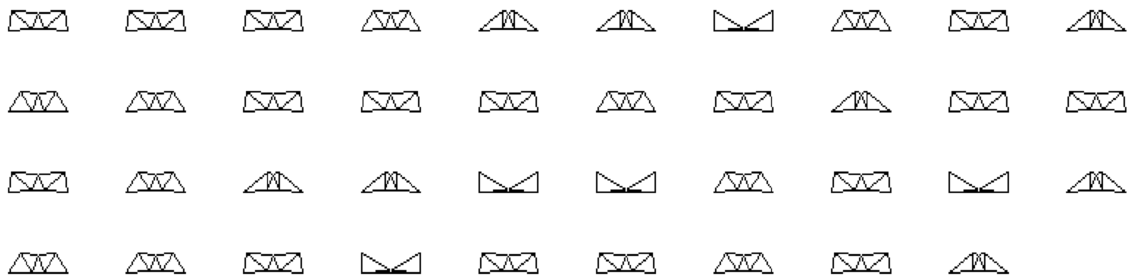
Shrestha, S.M. and Ghaboussi, J. (1998). “Evolution of optimum structural shapes using genetic algorithm.” *Journal of Structural Engineering*, 124(8), 1331-1338.

Srinivas, N., and Deb, K. (1994). “Multiobjective optimization using nondominated sorting in genetic algorithms.” *Evolutionary Computation, 2*(3), 221-248.

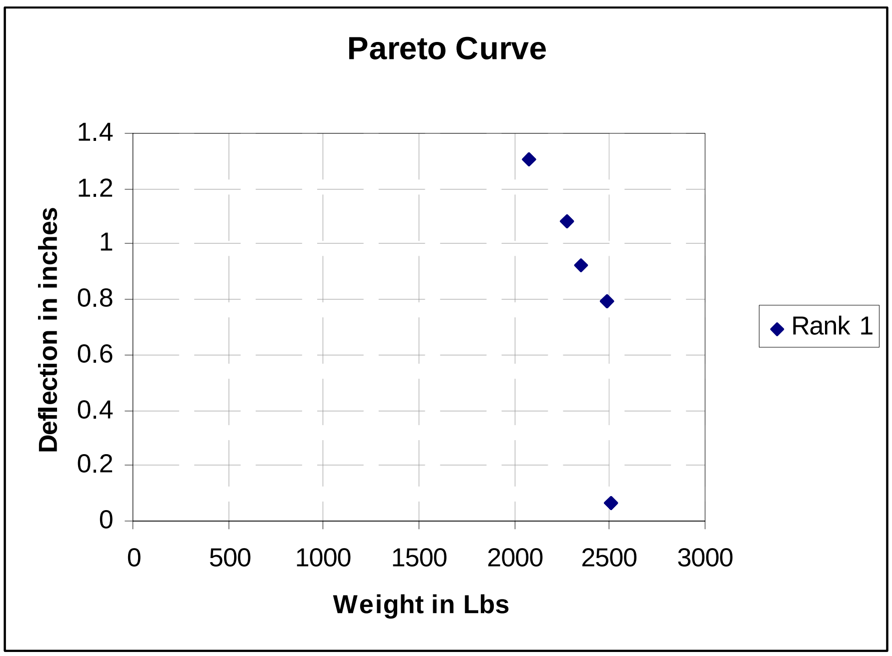
Yates, D. F., and Templeman, A. B. (1982). “The complexity of procedures for determining minimum weight trusses with discrete member sizes.” *Computers and Structures, 18 (1982)*, pp. 487{495).

Yeh, I.C. (1999). “Hybrid genetic algorithm for optimization of truss structures.” *Computer-Aided Civil and Infrastructure Engineering*, 14, 199-206.

APPENDIX I



Most optimal individuals after 1000 generations for 40 ft span (Trial 4)



Pareto curve showing the trade off between weight and deflection

Truss 1: Information

No. of elements	No. of nodes	Wt. of the truss (Lbs)	Deflection produced(inches)
7	11	2486.296143	0.793367

Nodal Information

Node X	Node Y
0	0
2	11
16	1
20	10
24	1
38	11
40	0

Nodal Connectivity

Nodal connectivity to form members	
1	2
1	3
2	3
2	4
3	4
6	7
7	5
6	5
6	4
5	4
3	5

Truss 4: Information

No. of elements	No. of nodes	Wt. of the truss	Deflection produced
7	11	2275.863281	1.080930

Nodal Information

Node X	Node Y
0	0
8	11

16	1
20	10
24	1
32	11
40	0

Nodal connectivity

Nodal connectivity to form members	
1	2
1	3
2	3
2	4
3	4
6	7
7	5
6	5
6	4
5	4
3	5

Truss 5: Information

No. of elements	No. of nodes	Wt. of the truss	Deflection produced
7	11	2073.486328	1.305574

Nodal Information

Node X	Node Y
0	0
16	1
16	11
20	10
24	11
24	1
40	0

Nodal connectivity

Nodal connectivity to form members	
1	2
1	3
2	3
2	4
3	4
6	7
7	5
6	5
6	4
5	4
2	6

Truss 7: Information

No. of elements	No. of nodes	Wt. of the truss	Deflection produced
5	7	2503.818359	0.061649

Nodal Information

Node X	Node Y
0	0
0	11
20	1
40	11
40	0

Nodal Connectivity

Nodal connectivity to form members	
1	2
1	3
2	3
4	5
5	3
4	3
1	5

Truss 10: Information

No. of elements	No. of nodes	Wt. of the truss (Lbs)	Deflection produced(inches)
7	11	2073.486328	1.305574

Nodal Information

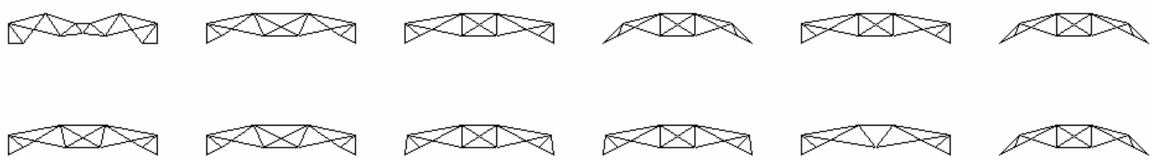
Node X	Node Y
0	0
16	1
16	11
20	10
24	11
24	1
40	0

Member Connectivity

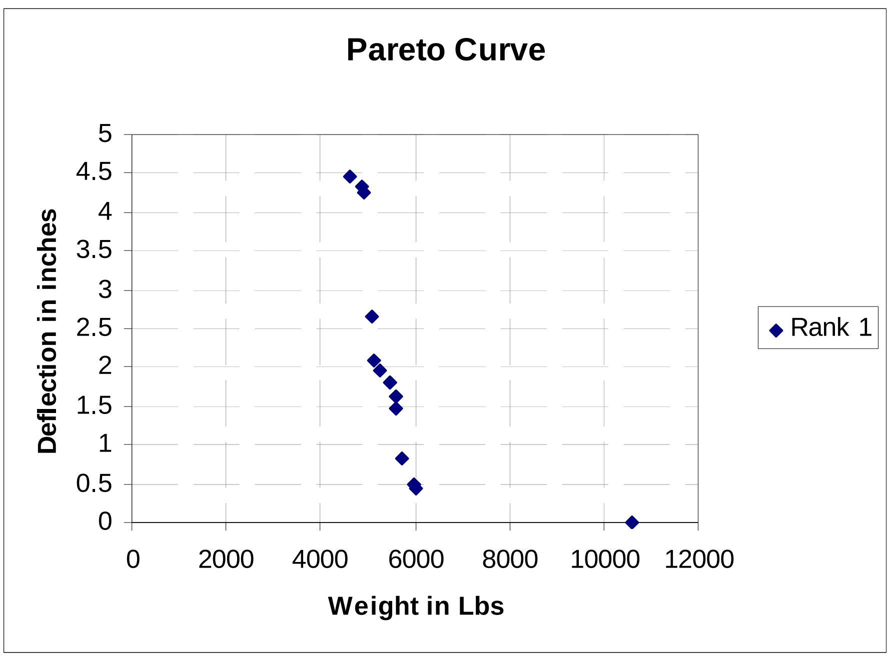
Nodal connectivity to form members	
1	2
1	3
2	3
2	4
3	4
6	7
7	5
6	5
6	4
5	4
2	6

All the other trusses are just a copy of the above trusses.

APPENDIX I I



Most optimal individuals after 1000 generations for 60 ft span (Trial 4)



Pareto curve showing the trade off between weight and deflection

Truss 1: Information

No. of elements	No. of nodes	Wt. of the truss (Lbs)	Deflection produced(inches)
13	23	5713.537	0.83416

Nodal Information

Node X	Node Y
0	0
6	0
0	8
15	12
21	3
27	8
30	4
33	8
39	3
45	12
60	8
54	0
60	0

Nodal Connectivity

Nodal connectivity to form members	
1	2
1	3
2	3
2	4
3	4
3	5
4	5
4	6
5	6
5	7
6	7
12	13
13	11
12	11
12	10
11	10
11	9
10	9
10	8
9	8
9	7
8	7

6	8
1	2

Truss 2: Information

No. of elements	No. of nodes	Wt. of the truss	Deflection produced
10	18	5588.617	1.474557

Nodal Information

Node X	Node Y
0	0
6	3
0	8
19	12
23	3
37	3
41	12
60	8
54	3
60	0

Nodal Connectivity

Nodal connectivity to form members	
1	2
1	3
2	3
2	4
3	4
3	5
4	5
4	6
4	7
9	10
10	8
9	8
9	7
8	7
8	6
7	6
5	7
5	6

Truss 4: Information

No. of elements	No. of nodes	Wt. of the truss	Deflection produced
10	18	4886.308	4.326453

Nodal Information

Node X	Node Y
0	0
6	3
8	8
23	12
23	3
37	3
37	12
52	8
54	3
60	0

Nodal Connectivity

Nodal connectivity to form members	
1	2
1	3
2	3
2	4
3	4
3	5
4	5
4	6
4	7
9	10
10	8
9	8
9	7
8	7
8	6
7	6
5	7
5	6

Truss 5

No. of elements	No. of nodes	Wt. of the truss	Deflection produced
10	18	5122.426	2.08865

Nodal Information

Node X	Node Y
0	0
6	3
0	8
23	12
23	3
37	3
37	12
60	8
54	3
60	0

Nodal Connectivity

Nodal connectivity to form members	
1	2
1	3
2	3
2	4
3	4
3	5
4	5
4	6
4	7
9	10
10	8
9	8
9	7
8	7
8	6
7	6
5	7
5	6

All the other trusses are just a copy of the above trusses.

VITA

Name: Soumya Sundar Pathi

Address: Plot J-11 Lewis Road,
Bhubaneswar Orissa, India. 751009

Email Address: soumyasundar_pathi@yahoo.com

Education: B.Tech, Civil Engineering, (National Institute of Technology Calicut), India. M.S., Civil Engineering Texas A&M University